# Propagation Techniques in Probabilistic Expert Systems

1997

Ross Morgan Curds

PhD Thesis

University College London

## Abstract

Techniques for the construction of probabilistic expert systems comprising both discrete and continuous random variables are presented. In particular we are concerned with how continuous random variables may be incorporated into an expert system - an area which has previously received relatively little attention. We investigate and extend the numeric techniques of other authors, and develop two new approaches. The first approach makes use of computer algebra. This exact technique enables a probability distribution to be expressed and manipulated in terms of its algebraic formula resulting in no loss of information.

Our second approach is an approximate method based upon cubic spline interpolation. We constrain the probability density function of a continuous variable to a finite set of points at which we have both function values and first derivatives. These values may then be held in a potential table and treated in an almost identical fashion to discrete variables. While symbolic techniques are shown to be only appropriate in special cases, cubic spline interpolation, though less accurate, is widely applicable.

We combine these techniques to form a hybrid methodology in which discrete variables, symbolic continuous variables, and spline interpolated continuous variables may exist not only in the same junction tree, but also in the same universe. We show how propagation algorithms may be constructed for these various cases and investigate how the means, variances and probability density functions of the marginal distributions in the system may be generated. It is shown how evidence of either a numeric or a symbolic nature may be incorporated into such systems and how simulation studies may be performed. The techniques we develop are implemented in the computer language *Mathematica* and an outline of how this may be accomplished is presented.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

The work described here considers a variety of techniques useful in the construction of *Probabilistic Expert Systems*. Ng & Abramson (1990) define an *Expert System* to be "a computer program which makes reasoned judgements that are at least on a par with a human expert, using a database that has been designed with the help of a human expert to tackle a particular, and often narrow, problem. Such programs are designed to be used when a human expert is not available, or to assist when the complexity of the problem at hand may make him fallible". If the problem at hand contains some form of uncertainty it will make sense to represent this uncertainty in a probabilistic way. We will then need to consider which random variables govern our problem, how may we represent them, and how do they interact with each other? With the addition of probability theory to our model it will be quite flexible in structure, but this increased flexibility may have a trade off in increased complexity. If the model comprises a large number of variables, has a large number of interactions, or if the random variables take complex probability distributions, then the model is likely to become too unwieldy for a human expert to manage without the aid of a computer. Computers bring their own problems to bear on our model. While they may be faster, more reliable and more accurate than their human counterpart, their approach to a problem is entirely dependent upon their programming. We will therefore require some structured way in which to build our model. A *probabilistic expert system* (*PES*) provides an appropriate framework on which to construct our methodology. A PES is an expert system which provides a method for the specification and handling of the joint distribution of a finite set of random variables (Lauritzen & Spiegelhalter, 1988).

## 1.1 Probabilistic Expert Systems

In Chapter 2 we will review the current theory surrounding PESs where this theory will be a necessary requirement for the development of our new methodology. We will specify a PES as a graph consisting of a finite set of nodes and a finite set of edges. Each node will correspond to a random variable in our model and each edge will represent some form of relationship between the two variables it joins. A directed edge between two variables will be used to represent a causal relationship in that the level of one variable will be assumed to have been *caused* by the level of the other. Similarly, an undirected edge will be used to represent some form of symmetric relationship between the two variables it joins. We will assume that the variables and their connections have been predefined by some "expert" and we will therefore not be concerned with the details of their construction. The network formed by the nodes and edges will be shown to fully express the conditional independence properties (or *Markov properties*) of the random variables in the model. It will also provide a way in which the joint distribution of the random variables in our model may be factorised. We will consider networks consisting of directed edges (*directed graphs*), undirected edges (*undirected graphs*) or a mixture of both (*chain graphs*). The random variables underlying our model may be a set of discrete random variables, a set of continuous random variables, or a set of both discrete and continuous variables. The graphs corresponding to these variable sets will be termed *discrete*, *continuous* and *mixed* respectively. We will, in the main, concentrate on problems concerning *mixed directed graphs*, however, most of the techniques we develop may be applied to any type of graph.

We will show how an undirected graph may be derived from a directed or chain graph such that the original graph possesses the Markov properties of its associated undirected graph. We will then be able to show how an undirected graph, and hence directed and chain graphs, may be compiled to form a higher level structure termed a *junction tree*. A junction tree will consist of a finite set of nodes termed *cliques* joined by a finite set of undirected edges termed *separators*. Associated with the cliques and separators there will be sets of random variables derived from the associated undirected graph and *potential functions* derived from the probability distributions of these random variables. We will term a clique or a separator a *universe*. We will show how a junction tree retains all the conditional independence properties of the underlying undirected graph. A junction tree will provide a simple structured way in which we may deal with our random variables.

We will show how a *propagation algorithm*, which utilises the properties of the junction tree, can be employed to set the potential function of every universe in the tree equal to the joint probability density function of the variables in that universe. We will also show how we may update these probability distributions given evidence on a collection of the random variables.

## 1.2 Numeric Techniques

Chapter 3 discusses the use of *numeric techniques*. We consider the discrete exact case which is the archetype of a PES. It consists of discrete random variables only. These are defined by a list of possible configurations and their associated weights. We introduce a class of distributions termed *Conditional Gaussian distributions* (Lauritzen & Wermuth, 1989), which possess properties which make them particularly suitable for use in PESs. Lauritzen (1992) presents a scheme which enables these distributions to be modelled and allows for the derivation of the probabilities, means and variances of the marginal distributions of the random variables. Implementations of this methodology have been made for *HUGIN* (Olesen, 1991) and *CAPRES* (Gammerman *et al.*, 1991). We will present and prove Lauritzen's methodology and extend it to show how simulation exercises may be carried out. We will use this scheme as an introduction to the construction of PESs. The language we will use will be *Mathematica*. This is a very versatile language which has several advantageous features making it an ideal development environment. These features include a convenient data structure, numeric and graphical capabilities and in particular the ability to handle symbolic expressions. It is this latter feature which we will find particularly useful later on. Chapter 3 introduces Lauritzen's "waste incinerator problem". This theoretical example seeks to model the processes occurring in a waste incinerator plant. The model comprises a set of multinomial discrete variables and conditionally normal continuous variables. We will be using this model to demonstrate the techniques we develop in later chapters and to compare the results of using our different methods.

## 1.3 Symbolic Techniques

In Chapter 4 we attempt to improve the amount of information we can gain from our model by taking a different approach to Lauritzen (1992). Instead of using nu-

meric techniques to compute probabilities, means and variances we seek to develop a methodology by which computer algebra can be used to manipulate probability distributions directly through their formulae. We will collectively term these methods *symbolic techniques*. We devise a *symbolic propagation algorithm*. This is an extension of the standard, numeric, propagation algorithm which uses computer algebra to propagate potential functions in the form of algebraic formulae. It results in no loss of information. Any statistics or graphs required may then be deduced from the potential functions following propagation. A by-product of this approach is the fact that *symbolic evidence* may be entered into the system. In other words a dummy variable may be entered as evidence on some continuous random variable and then be propagated throughout the system. Statistics and probability distributions relating to the other random variables will then be expressed as functions of the dummy variable. This will enable us to determine the effect of evidence on the system more easily. We present two purely discrete networks and show how symbolic techniques may be used to model them.

## 1.4 Spline Approximation Techniques

A drawback of symbolic techniques is that they will only work in special cases. In models which contain continuous random variables we will require every necessary marginalisation with respect to a continuous random variable to be expressible in terms of an integral which may be solved in closed form. Of course even if the integral actually is solvable in closed form it will need to be one which our system recognises. These generalities will be difficult to fulfil and we will often be forced to resort to approximation techniques. In Chapter 5 we present a method for the cubic spline approximation of the probability density functions of continuous random variables and show how such approximations may be used to build probabilistic expert systems consisting of both discrete and continuous random variables. We show how the cubic spline functions we form can be represented in a simple tabular form consisting of function values and first derivatives, and how these tables may be treated in an almost identical way to discrete variables. We demonstrate how these representations may be propagated throughout a junction tree and show how the means, variances, probabilities, and marginal probability density functions of the random variables concerned may be generated. In addition we consider how we may display our results graphically. We will term these methods *spline approximation techniques*.

## 1.5 Hybrid Techniques

In Chapter 6 we seek to unite the methodology developed in previous chapters by describing how *hybrid techniques* may be used to mix both symbolic and spline approximation techniques in the same model. The data structures required for each technique blend together in a natural fashion allowing hybrid universes to be built in contrast to the scheme described by Dawid, Kjærulff, & Lauritzen (1993) which allows for hybrid trees but non-hybrid universes. We describe how a *hybrid propagation scheme* may be set up for a probabilistic expert system in which the discrete random variables are assumed to be multinomially distributed and may have no continuous parents. The continuous variables are partitioned into spline interpolated variables and symbolic variables. The methods we develop can in theory, by suitable partitioning of the continuous variables, be applied to any mixed PES given the constraints on the discrete variables. Despite this implied flexibility a generally robust partitioning is difficult to provide. We thus illustrate our methodology by one possible partitioning based on the existence of a set of conditional Gaussian distributions. The usual techniques for the addition of evidence and the construction of means, variances, probabilities, marginal probability density functions, and graphical representations are also provided.

## 1.6 Conclusions

The final chapter in this work summarises our conclusions on the material presented and considers what additional theory should be developed in the future to improve and extend our results.

# Chapter 2

# Probabilistic Expert Systems

## 2.1 Introduction

This introductory chapter seeks to review some of the current theory surrounding the construction of *probabilistic expert systems* (PESs) and as such much of it is derived from standard text. The theory we shall present, however, will attempt to provide enough background information to enable the development of our new methodology. We will start by defining conditional independence and reviewing graph theory - these concepts will give us the building blocks we require for PES construction. We will specify a PES for a model as a graph consisting of a finite set of nodes and a finite set of edges. Each node will correspond to a random variable in our model and each edge, whether directed or undirected, will represent some form of relationship between the two nodes it joins. We will assume that the variables and their connections have been predefined by some "expert" and we will therefore not be too concerned with the details of their construction. We will show that the network formed by the nodes and edges fully expresses the conditional independence properties (or *Markov properties*) of the random variables in the model and provides a way in which their joint distribution may be factorised.

The networks we will consider will consist of directed edges (*directed graphs*), undirected edges (*undirected graphs*) or a mixture of both (*chain graphs*). The random variables underlying our model may be a set of discrete random variables (*discrete graphs*), a set of continuous random variables (*continuous graphs*), or a set of both discrete and continuous variables (*mixed graphs*). We will show how an undirected graph may be derived from both directed and chain graphs such that the Markov properties of the original graph are contained in that undirected graph. We will then show how an undirected graph, and hence any graph, may be

compiled to form a higher level structure termed a *junction tree*. We will show how a junction tree retains the Markov properties of the underlying undirected graph. A junction tree will provide a simple structured way in which we may deal with our random variables. We will show how a *propagation algorithm*, which utilises the properties of the junction tree, can be employed to set the potential function of every universe in the tree equal to the joint probability density function of the variables in that universe. We will also show how we may update these probability distributions given evidence on a collection of the random variables.

## 2.2 Conditional Independence

We begin by considering how a set of random variables may interact. What we are seeking is some concept which will enable us to simplify the modelling of the joint distribution of a set of random variables by partitioning that joint distribution into a series of simpler distributions each of which will be easier to handle than the joint distribution of the entire set. We will introduce some key concepts in this area. Let us first define $X$ and $Y$ to be random variables (possibly vectors) and let $f_{X,Y}(x,y)$ denote the joint probability density of $X$ and $Y$. If $X$ and $Y$ are both discrete then $f_{X,Y}(x,y)$ denotes the joint probability function $P(X=x, Y=y)$. We may define a property termed *independence* as follows:

**Definition 1 Independence:** *$X$ and $Y$ are said to be independent if and only if:*

$$f_{X,Y}(x,y) = f_X(x) f_Y(y) \qquad for\ all\ x, y.$$

*This may be denoted $X \perp\!\!\!\perp Y$.*

*Equivalently:*

$$X \perp\!\!\!\perp Y \Leftrightarrow f_{X|Y}(x;y) = f_X(x) \qquad for\ all\ x; for\ all\ y\ such\ that\ f_Y(y) > 0$$

*where $f_{X|Y}(x;y)$ is the conditional distribution of $X$ given $Y$ i.e. $f_{X|Y}(x;y) = f_{X,Y}(x,y)/f_Y(y)$.*

In order to determine whether variables are independent or not we need only determine whether their joint density factorises or not. This is made clear in the following proposition.

**Proposition 1 Factorisation Criterion for Independence:** *The random vectors $X$ and $Y$ are independent if and only if there exist two functions $g$ and $h$ such that:*

$$f_{X,Y}(x,y) = g(x)h(y) \qquad for\ all\ x, y.$$

**Proof** The implication is by the definition of independence (Definition 1) with $g = f_X$, $h = f_Y$. The converse is proved by marginalising with respect to $x$ and $y$ separately giving:

$$f_Y(y) \propto h(y) \qquad f_X(x) \propto g(x)$$

Thus:

$$f_{X,Y}(x,y) = g(x)h(y) \propto f_X(x)f_Y(y)$$

Marginalising with respect to both $x$ and $y$ gives $f_{X,Y}(x,y) = f_X(x)f_Y(y)$.

□

We also introduce the concept of *reduction* as follows:

**Proposition 2 Reduction:** *Joint independence implies marginal independence (but not conversely). If $(X,Y,Z)$ is a partitioned random vector then:*

$$X \perp\!\!\!\perp (Y,Z) \Rightarrow X \perp\!\!\!\perp Y \ and \ X \perp\!\!\!\perp Z.$$

**Proof** $X \perp\!\!\!\perp (Y,Z) \Rightarrow f_{X,Y,Z}(x,y,z) = f_X(x)f_{Y,Z}(y,z)$ by definition. Marginalising with respect to $Z$ gives:

$$f_{X,Y}(x,y) = f_X(x)f_Y(y) \qquad \text{i.e. } X \perp\!\!\!\perp Y$$

And similarly marginalising with respect to $Y$ gives:

$$f_{X,Z}(x,z) = f_X(x)f_Z(z) \qquad \text{i.e. } X \perp\!\!\!\perp Z$$

□

Related to the concept of independence is that of *conditional independence* which is defined as follows:

**Definition 2 Conditional Independence:** *$X$ and $Y$ are said to be conditionally independent given $Z$, if and only if:*

$$f_{X,Y|Z}(x,y;z) = f_{X|Z}(x;z)f_{Y|Z}(y;z)$$

*for all $x$, $y$; for all $z$ such that $f_Z(z) > 0$. This is written as $X \perp\!\!\!\perp Y \mid Z$.*

*Equivalently, for all $x$, $y$; for all $z$ such that $f_Z(z) > 0$:*

$$X \perp\!\!\!\perp Y \mid Z \Leftrightarrow f_{X|Y,Z}(x; y, z) = f_{X|Z}(x; z)$$

$$X \perp\!\!\!\perp Y \mid Z \Leftrightarrow f_{X,Y,Z}(x, y, z) = \frac{f_{X,Z}(x, z) f_{Y,Z}(y, z)}{f_Z(z)}$$

We then have the following propositions for conditional independence:

**Proposition 3 Factorisation Criterion for Conditional Independence:** *The random vectors $X$ and $Y$ are conditionally independent given $Z$, if and only if there exist functions $g$, $h$ such that:*

$$f_{X,Y,Z}(x, y, z) = g(x, z)h(y, z) \qquad \text{for all } x, y; \text{for all } z \text{ such that } f_Z(z) > 0$$

**Proof** The implication is by the definition of conditional independence (Definition 2) with $g = f_{X|Z}$ and $h = f_{Y|Z} f_Z$ or $g = f_{X|Z} f_Z$ and $h = f_{Y|Z}$. The converse may be proved by marginalising with respect to $x$ and $y$ separately giving:

$$f_{Y,Z}(y, z) \propto g'(z)h(y, z) \qquad f_{X,Z}(x, z) \propto g(x, z)h'(z)$$

Thus:

$$f_{X,Y,Z}(x, y, z) = g(x, z)h(y, z) \propto \frac{f_{X,Z}(x, z) f_{Y,Z}(y, z)}{g'(z)h'(z)}$$

Marginalising with respect to both $x$ and $y$ gives:

$$f_{X,Y,Z}(x, y, z) = \frac{f_{X,Z}(x, z) f_{Y,Z}(y, z)}{f_Z(z)}$$

□

**Proposition 4 Reduction:** *If $(X, Y_1, Y_2, Z)$ is a partitioned random vector then:*

$$X \perp\!\!\!\perp (Y_1, Y_2) \mid Z \Rightarrow X \perp\!\!\!\perp Y_1 \mid Z \text{ and } X \perp\!\!\!\perp Y_2 \mid Z$$

**Proof** $X \perp\!\!\!\perp (Y_1, Y_2) \mid Z \Rightarrow f_{X,Y_1,Y_2|Z}(x, y_1, y_2; z) = f_{X|Z}(x; z) f_{Y_1,Y_2|Z}(y_1, y_2; z)$ by definition. Marginalising with respect to $Y_2$ gives:

$$f_{X,Y_1|Z}(x, y_1; z) = f_{X|Z}(x; z) f_{Y_1|Z}(y_1; z) \qquad \text{i.e. } X \perp\!\!\!\perp Y_1 \mid Z$$

And similarly marginalising with respect to $Y_1$ gives:

$$f_{X,Y_2|Z}(x, y_2; z) = f_{X|Z}(x; z) f_{Y_2|Z}(y_2; z) \qquad \text{i.e. } X \perp\!\!\!\perp Y_2 \mid Z$$

□

**Proposition 5 Block Independence:** *If* $(X, Y_1, Y_2, Z)$ *is a partitioned random vector and* $f_{X,Y_1,Y_2,Z}$ *is positive then:*

$$X \perp\!\!\!\perp (Y_1, Y_2) \mid Z \Leftrightarrow X \perp\!\!\!\perp Y_1 \mid (Y_2, Z) \text{ and } X \perp\!\!\!\perp Y_2 \mid (Y_1, Z)$$

**Proof** $X \perp\!\!\!\perp (Y_1, Y_2) \mid Z \Rightarrow$

$$\begin{aligned} f_{X,Y_1,Y_2,Z}(x, y_1, y_2, z) &= g_0(x, z) h_0(y_1, y_2, z) \\ &= g_1(x, y_2, z) h_0(y_1, y_2, z) \quad \Rightarrow \quad X \perp\!\!\!\perp Y_1 \mid (Y_2, Z) \\ &= g_2(x, y_1, z) h_0(y_1, y_2, z) \quad \Rightarrow \quad X \perp\!\!\!\perp Y_2 \mid (Y_1, Z) \end{aligned}$$

Conversely $X \perp\!\!\!\perp Y_1 \mid (Y_2, Z) \Rightarrow$

$$f_{X,Y_1,Y_2,Z}(x, y_1, y_2, z) = g_1(x, y_2, z) h_0(y_1, y_2, z)$$

and $X \perp\!\!\!\perp Y_2 \mid (Y_1, Z) \Rightarrow$

$$g_1(x, y_2, z) = g_3(x, z) g_4(y_2, z)$$

$$\begin{aligned} \Rightarrow \quad f_{X,Y_1,Y_2,Z}(x, y_1, y_2, z) &= g_3(x, z) g_4(y_2, z) h_0(y_1, y_2, z) \\ &= g_3(x, z) h_1(y_1, y_2, z) \\ \Rightarrow \quad & X \perp\!\!\!\perp (Y_1, Y_2) \mid Z \end{aligned}$$

□

Putting block independence and reduction together we get:

$$\begin{matrix} X \perp\!\!\!\perp Y_1 \mid (Y_2, Z) \\ X \perp\!\!\!\perp Y_2 \mid (Y_1, Z) \end{matrix} \Leftrightarrow X \perp\!\!\!\perp (Y_1, Y_2) \mid Z \Rightarrow \begin{matrix} X \perp\!\!\!\perp Y_1 \mid Z \\ X \perp\!\!\!\perp Y_2 \mid Z \end{matrix}$$

Thus:

$$\begin{matrix} X \perp\!\!\!\perp Y_1 \mid (Y_2, Z) \\ X \perp\!\!\!\perp Y_2 \mid (Y_1, Z) \end{matrix} \Rightarrow \begin{matrix} X \perp\!\!\!\perp Y_1 \mid Z \\ X \perp\!\!\!\perp Y_2 \mid Z \end{matrix}$$

and:

$$X \perp\!\!\!\perp (Y_1, Y_2) \mid Z \Rightarrow \begin{matrix} X \perp\!\!\!\perp Y_1 \mid Z \\ X \perp\!\!\!\perp Y_2 \mid (Y_1, Z) \end{matrix}$$

Thus it is clear that if we use the conditional independence properties of a set of random variables we may factorise their joint distribution into a series of simpler distributions. This gives us an insight into how we might proceed to model a complex system more conveniently. Further discussions of conditional independence may be found in Dawid (1979a, 1979b).

## 2.3 Graph Theory

In this section we will review the notation and basics of *graph theory*. Graph theory provides both a structured way in which we may represent a complex system and, as we will discover, a structured way in which we may tackle the modelling of that system.

A *graph* $\mathcal{G}$ is defined as a pair $(K, E)$, where $K$ represents a finite set of *vertices* and the *edges* $E$ are a subset of the set $K \times K$ of ordered pairs of distinct vertices. The graphs we will be concerned with are termed *simple* since they may contain no multiple edges or loops. If a pair $(a, b) \in E$ and $(b, a) \in E$ then we have an *undirected edge*, or *line* , between $a$ and $b$. This is denoted $a \sim b$. If, however, $(a, b) \in E$ but $(b, a) \notin E$ then we have a *directed edge*, or *arrow*, between $a$ and $b$. This is denoted $a \rightarrow b$. Special types of graphs are defined by the relationships contained within them. An *undirected graph*, for example, is one which has only undirected edges while a *directed graph* has only directed edges. Figure 2.1 shows a directed graph, Figure 2.2 shows an undirected graph, and Figure 2.3 shows a graph which contains both directed and undirected edges. All three graphs have the same vertex set $K = \{1, 2, 3, 4, 5\}$ and have relationships between the same variables yet these relationships are different for the three types of graph, since they have different edge sets. They therefore have different meanings.

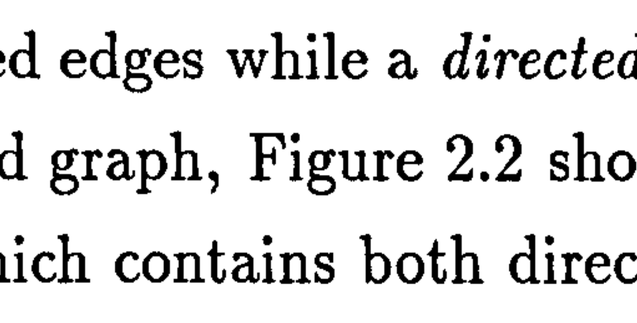



Figure 2.1: A directed graph.

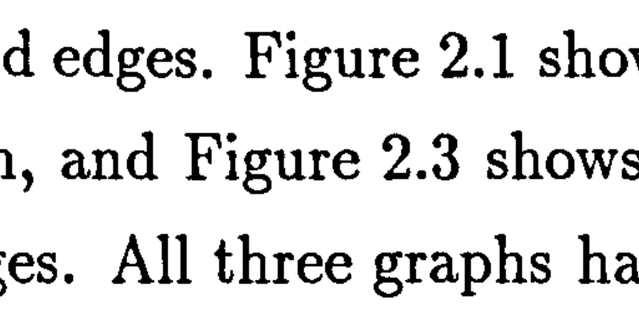



Figure 2.2: An undirected graph.




Figure 2.3: A graph containing both directed and undirected edges.

The *symmetrisation* $\mathcal{G}^{\sim}$ of a graph $\mathcal{G}$ represents the undirected graph corresponding to the graph $\mathcal{G}$. This may be obtained from $\mathcal{G}$ by substituting lines for all the existing arrows. We can therefore see that Figure 2.2 represents the symmetrisation of both Figures 2.1 and 2.3 and, trivially, of itself. If $A \subseteq K$ is a subset of the vertex set then it induces a *subgraph* $\mathcal{G}^A = (A, E_A)$, with edge set $E_A = E \cap (A \times A)$. In other words the induced subgraph is obtained from $\mathcal{G}$ by keeping the vertices in $A$ and the edges with both endpoints belonging to $A$.

A *complete graph* is one in which every vertex is joined to every other vertex by either an arrow or a line. Thus the symmetrisation of a complete graph with $n$ vertices has $\binom{n}{2}$ lines. A *complete subset* is a subset that induces a *complete subgraph* in the symmetrisation $\mathcal{G}^{\sim}$ of $\mathcal{G}$. In Figures 2.1, 2.2 and 2.3 the subsets $\{2,4\}$ and $\{4\}$ are just two of the complete subsets of $\{1,2,3,4,5\}$ the latter case being trivial. This enables us to define the concept of a *clique* which is a complete subset that is maximal with respect to the number of vertices it contains. $\{2,4\}$ is therefore a clique yet $\{4\}$ is not.

Relationships of vertices to one another may also be defined. A vertex $a$ is said to be a *parent* of $b$ if an arrow from $a$ points towards $b$. Similarly $b$ is then termed the *child* of $a$. Vertices $a$ and $b$ are said to be *adjacent* or *neighbours* if a line exists between $a$ and $b$. This is denoted by $a \sim b$. Hence if we consider Figure 2.3 we may note that 2 is a parent of both 3 and 5. 3 and 5 are children of 2 and 4 is a neighbour of 2. We also use the terms $pa(a)$ for the parents of $a$, $ch(a)$ for the children of $a$ and $ne(a)$ for the neighbours of $a$. Similar notation exists for subsets where we define the corresponding relationships for a given subset $A$ with members $a$ thus:

$$\begin{aligned} pa(A) &= \bigcup_{a \in A} pa(a) \backslash A \\ ch(A) &= \bigcup_{a \in A} ch(a) \backslash A \\ ne(A) &= \bigcup_{a \in A} ne(a) \backslash A \end{aligned}$$

This notation trivially applies for subsets with only a single member i.e. in the case of a single vertex. In the above we take the notation $K \backslash A$ to be the remaining vertices in the vertex set $K$ when the set of vertices $A$ is removed. The *boundary* of a subset $A$ is the set of vertices in $K \backslash A$ that are either parents or neighbours to vertices in $A$ i.e.

$$bd(A) = pa(A) \cup ne(A)$$

A related concept is the *closure* of $A$ which is defined:

$$cl(A) = A \cup bd(A)$$

and a subset $A$ is thus termed *closed* if $A = cl(A)$. If we consider the graph in Figure 2.3 and define a subset $S = \{2,5\}$ of $K$ then $pa(S) = \{1\}$, $ne(S) = \{4\}$, $ch(S) = \{3\}$. Also we have $bd(S) = \{1,4\}$ and $cl(S) = \{1,2,4,5\}$, so $S$ is not closed. Considering the trivial case of $K$ we notice that the boundary of $K$ is empty and hence $K$ is closed.

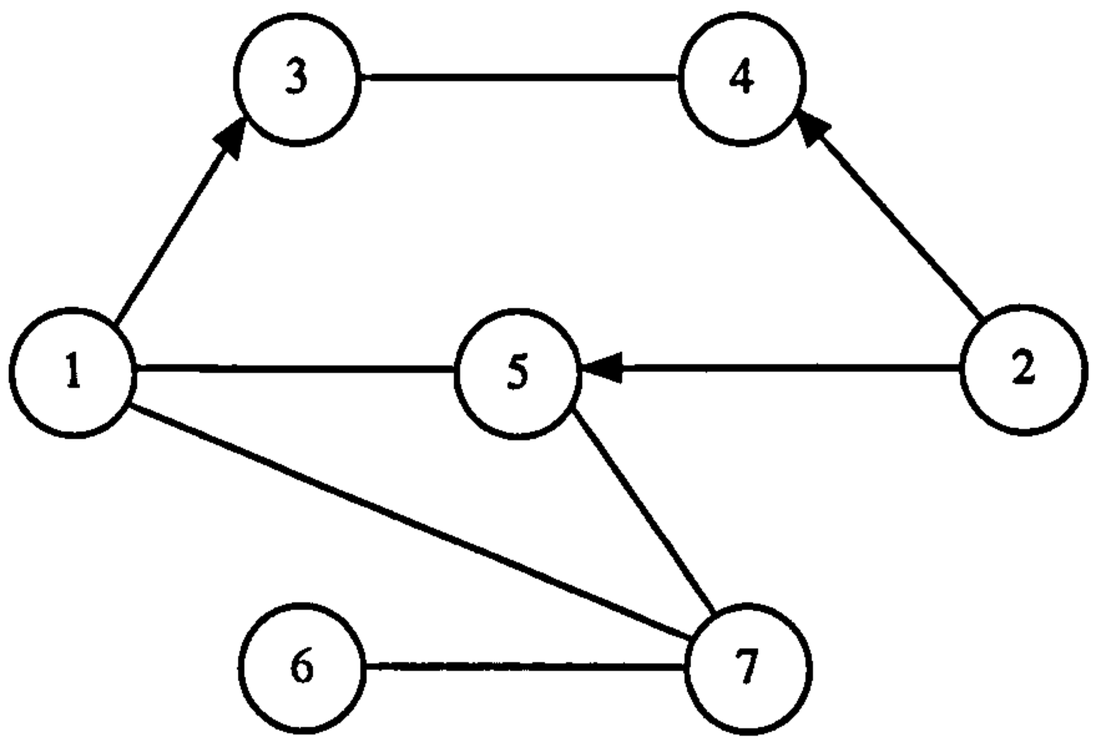



Figure 2.4: A second graph containing both directed and undirected edges.

A *path* of length $n$ from $a$ to $b$ is defined to be a sequence $a = a_0, a_1, \ldots, a_n = b$ of $n+1$ distinct vertices such that $(a_{i-1}, a_i) \in E$ for all $i = 1,2,\ldots,n$. A path is termed *short* if no proper subset of the path is also a path from $a$ to $b$. A vertex $a$ *leads* to a second vertex $b$ if a path exists from $a$ to $b$. This is denoted $a \mapsto b$. In Figure 2.4 $1 \mapsto 6$. This is facilitated through two possible paths, one of length two $(1,7,6)$, and one of length three $(1,5,7,6)$. The *ancestors* of $b$, $an(b)$, are the set of vertices $a \in E$ such that $a \mapsto b$. The *descendants* of $a$, $de(a)$, are the set of vertices $b \in E$ such that $a \mapsto b$. The *non-descendants* of $a$ are defined $nd(a) = K \backslash [de(a) \cup \{a\}]$. Thus in Figure 2.4 $an(1) = \{2,5,6,7\}$, $de(1) = \{3,4,5,6,7\}$ and $nd(1) = \{2\}$ . A set of vertices $A$ is said to be *ancestral* if $an(A) \subseteq A$. In Figure 2.4 the set $\{1,2,5,6,7\}$ is ancestral, and, trivially, so is the set $\{2\}$. The intersection $B$, say, of two ancestral sets is also ancestral assuming $B$ is well-defined.

Nodes $a$ and $b$ are said to *connect* if both $a \mapsto b$ and $b \mapsto a$, denoted $a \rightleftharpoons b$. Thus 1 and 6 connect in Figure 2.4, as do 1 and 5. The *connectivity component*

$co(a)$ of a graph $\mathcal{G}$ for $a \in K$ is the set of variables which connect to $a$. Thus $b \in co(a) \Leftrightarrow a \rightleftharpoons b$. In Figure 2.4 $co(1) = \{1,5,6,7\}$, $co(3) = \{3,4\}$, and $co(2) = \{2\}$. These connectivity components are illustrated in Figure 2.5. If $a \in A \subseteq K$ then $co(a)_A$ denotes the connectivity component of $a$ in $\mathcal{G}_A$. Hence if $A = \{2,3,4\}$ then $co(3)_A$ (or $co(4)_A$ ) denotes the connectivity component $\{3,4\}$ in $\mathcal{G}_A$.

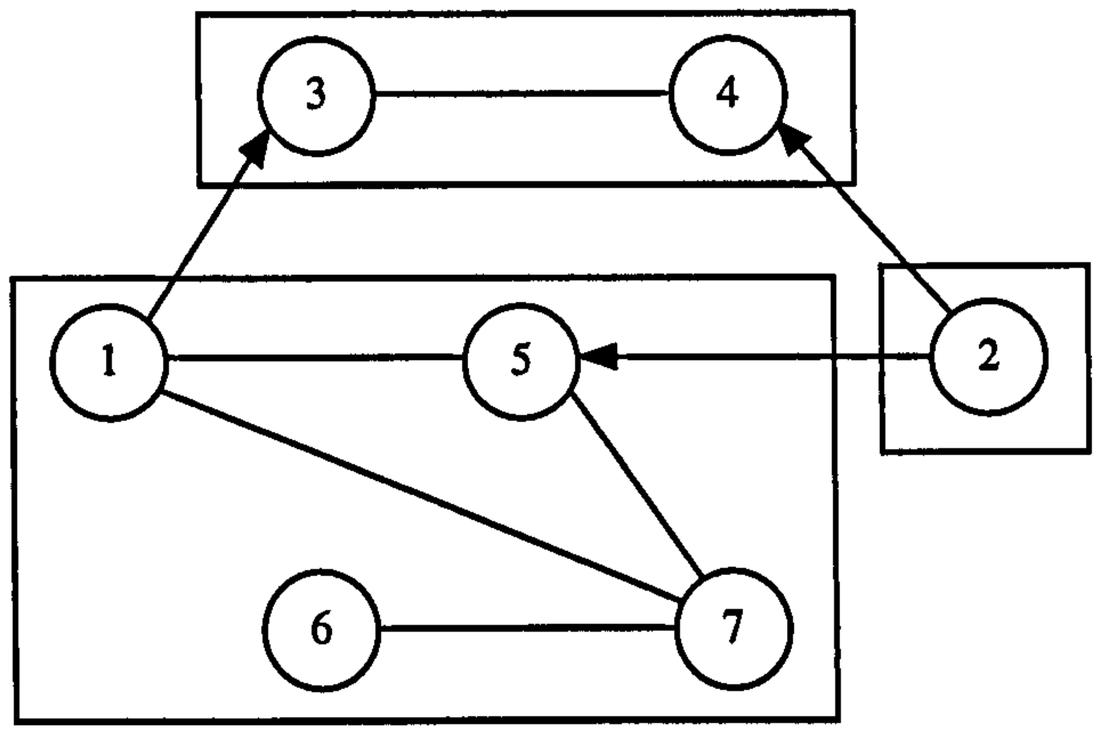



Figure 2.5: The connectivity components of the graph in Figure 2.4.

A *walk* of length $n$ from $a$ to $b$ is a sequence $a = a_0, a_1, \ldots, a_n = b$ of distinct vertices such that $a_{i-1} \rightarrow a_i$ or $a_i \rightarrow a_{i-1}$, for $i = 1,2,\ldots,n$. There are thus 3 possible walks from 1 to 2 in Figure 2.4. These are the sequences $(1,3,4,2)$, $(1,5,2)$ and $(1,7,5,2)$. A subset of vertices *separates* two vertices, $a$ and $b$, if every path joining the two vertices contains at least one vertex from the separating subset. A subset $S$ *separates* $A$ from $B$ if all walks from vertices $a \in A$ to $b \in B$ intersect $S$. In Figure 2.4 $\{1,5\}$ separates $\{6,7\}$ from $\{2,3,4\}$. An *n-cycle* is a path of length $n$ with the modification that $a = b$ i.e. it begins and ends at the same point. In Figure 2.4 the sequence $(1,5,7,1)$ is a 3-cycle. An $n$-cycle is said to be *directed* if it contains an arrow.

Further details of graph theory may be found in Lauritzen (1989).

## 2.4 Independence Graphs

We will now consider how a graph may be used to represent the conditional independence structure of a set of random variables. Let $X = (X_1, X_2, \ldots, X_k)$ denote a vector of $k$ random variables, let $K = \{1,2,\ldots,k\}$ denote a corresponding set of vertices, and let $E$ be an edge set. Then the graph $\mathcal{G} = (K,E)$ is a *conditional independence graph* or *independence graph* if there exists no edge between

two vertices whenever the corresponding pair of variables is independent given all the remaining variables which we will term the *rest*. We may use the shorthand notation $1 \perp\!\!\!\perp 2 \mid \{3,4\}$ for $X_1 \perp\!\!\!\perp X_2 \mid (X_3, X_4)$. The independence of $X_i$ and $X_j$ given the rest may thus be written $X_i \perp\!\!\!\perp X_j \mid rest$ or $i \perp\!\!\!\perp j \mid K \backslash \{i,j\}$. Whittaker (1990) makes the following definition:

**Definition 3** *The* conditional independence graph *of $X$ is the undirected graph $\mathcal{G} = (K, E)$ where $K = \{1, 2, \ldots, k\}$ and $(i,j)$is not in the edge set $E$ if and only if $X_i \perp\!\!\!\perp X_j \mid X_{K \backslash \{i,j\}}$.*

We have thus defined the set of graphs which may be used to represent the symmetric associations that exist between pairs of variables. Let us consider a density function:

$$f_X(x) = exp(u + x_1 + x_1x_2 + x_2x_3x_4) \tag{2.1}$$

for $X = (X_1, X_2, X_3, X_4)$, where $u$ is a constant which ensures that the density integrates to 1. Then, since $f_X(x) > 0$ we may apply the factorisation criterion (Proposition 3) which implies that:

$$X_1 \perp\!\!\!\perp X_4 \mid (X_2, X_3) \text{ and } X_1 \perp\!\!\!\perp X_3 \mid (X_2, X_4)$$

We thus derive the independence graph of Figure 2.6, with cliques $\{1,2\}$ and $\{2,3,4\}$, which clearly shows the relationships that exist between the four variables. It can be seen that 1 is neither adjacent to 3 nor 4 expressing the conditional independences implied.

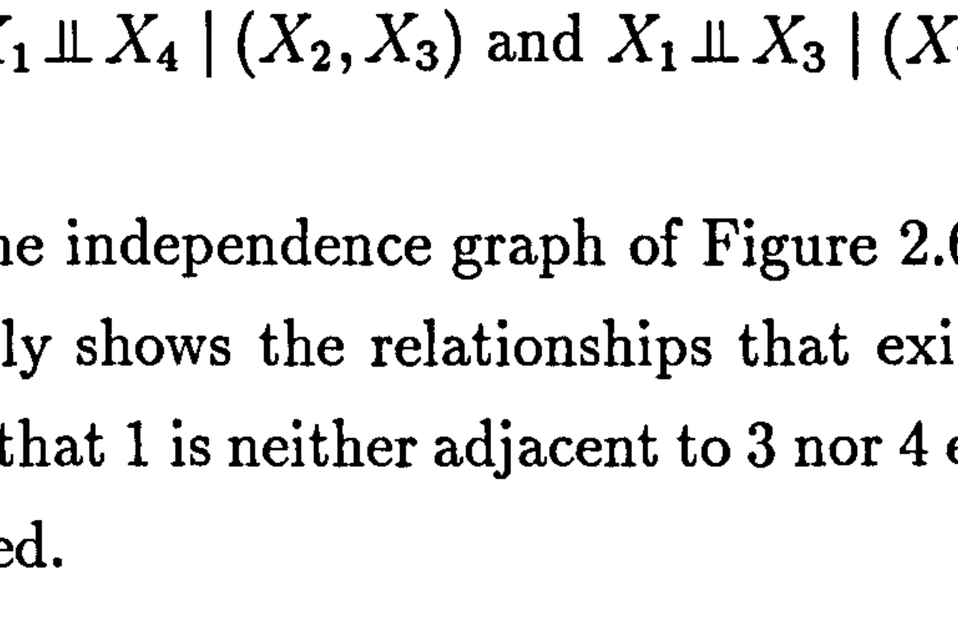



Figure 2.6: The independence graph for Equation 2.1.

## 2.5 Separation

Recall our definition of separation from above. Two vertices $i$ and $j$ are said to be *separated* by a subset, if every path joining the two vertices contains at least one vertex from the separating subset. The importance of separation is shown by the *separation theorem* which states that non-adjacent variables are independent given the separating set alone. We are thus able to break up our independence graph into subgraphs simplifying our problems. We introduce the following lemmas (see Whittaker, 1990) to enable us to prove the separation theorem for random variables with a positive joint probability density function.

**Lemma 1** *Suppose that the vertex set* $K = \{1, 2, \ldots, k\}$ *can be partitioned into two sets* $B$ *and* $C$ *where in the independence graph of* $K$ *there is no path between any vertex in* $C$ *with any vertex in* $B$. *Then:*

$$i \perp\!\!\!\perp j \mid A \text{ for all } i \in B \text{ and } j \in C$$

*for any subset* $A$ *of* $K$ *not containing* $i$ *or* $j$.

**Proof**

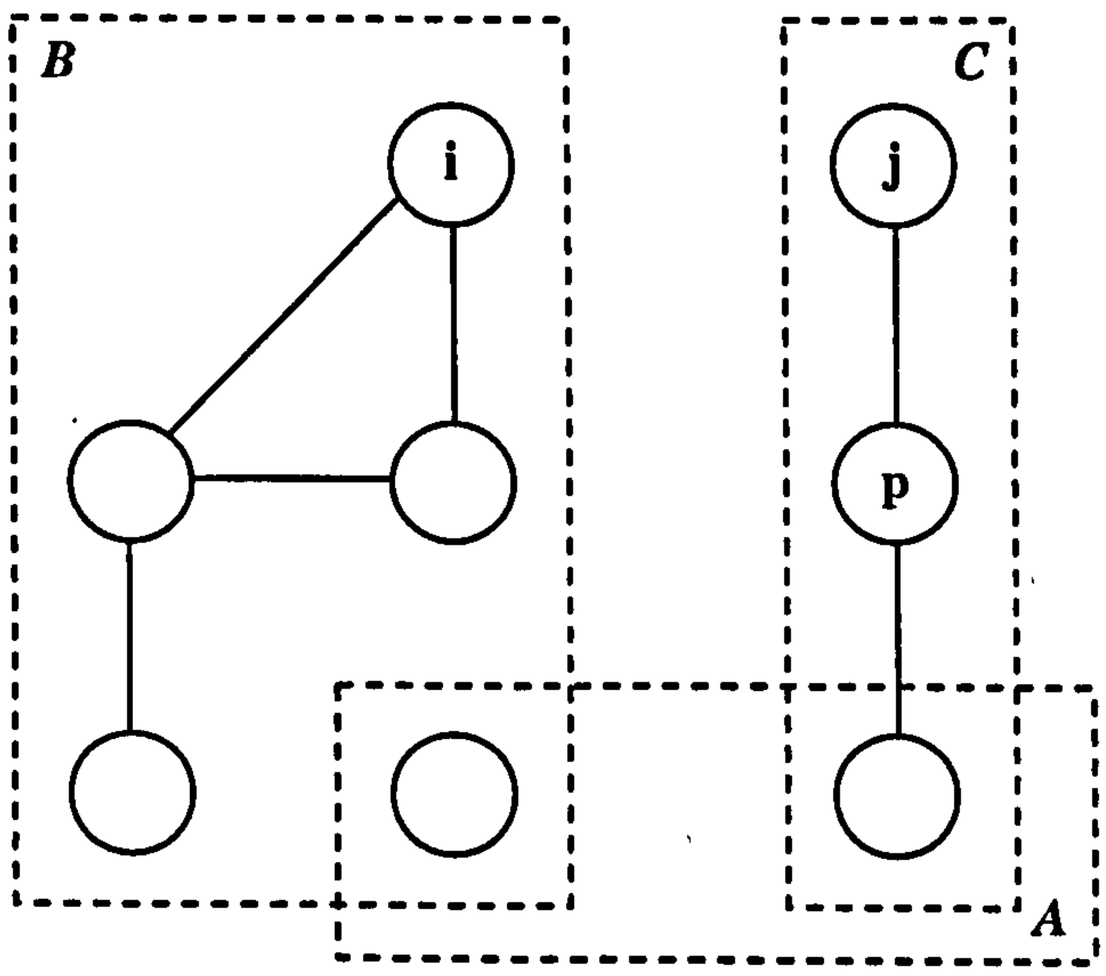



Figure 2.7: An independence graph to illustrate Lemma 1.

Consider Figure 2.7. Let us suppose two vertices $i$ and $j$ are not connected (i.e. not linked by a path) and let us select some arbitrary vertex $p$ in $C$ but not in $A \cup \{j\}$. By our construction of an independence graph we know that:

$$i \perp\!\!\!\perp j \mid K \backslash \{i, j\} \text{ and } i \perp\!\!\!\perp p \mid K \backslash \{i, p\}$$

Since the random variables possess a positive joint density we may apply block independence. It follows that:

$$i \perp\!\!\!\perp \{j,p\} \mid K\backslash\{i,j,p\}$$

and by reduction:

$$i \perp\!\!\!\perp j \mid K\backslash\{i,j,p\}$$

Thus we have eliminated vertex $p$ from our conditioning set. This independence statement will hold for all $i \in B$, $j \in C$ such that $j \neq p$. Thus the independence graph for the $k-1$ vertices $K\backslash\{p\}$ has no edge between any vertex in $B$ and any vertex in $C\backslash\{p\}$.

So, we may repeat this argument, at each stage removing a vertex $p \neq j$ in $C$ but not in $A \cup \{j\}$ until only those in $A \cup \{j\}$ remain. We may then proceed by removing vertices $p \neq i$ in $B$ but not in $A \cup \{i\}$ until only those in $A \cup \{i\}$ remain. We are left with $i \perp\!\!\!\perp j \mid A$, as required.

It should be noted that if $A$ were empty all the vertices except $i$ and $j$ would be removed and we would have $i \perp\!\!\!\perp j$.

□

**Lemma 2** *If $A$ is any subset of vertices of $K$ that separates two vertices $i$ and $j$ then $i \perp\!\!\!\perp j \mid A$, or more precisely, $X_i \perp\!\!\!\perp X_j \mid X_A$.*

**Proof** If $i$ and $j$ are not connected then the result follows immediately from Lemma 1. Consider, therefore, the case when $i$ and $j$ are connected. By hypothesis the vertices $i$ and $j$ are separated by $A$. The remaining vertices are either connected or not connected to $A \cup \{i,j\}$. If a vertex is connected to $A \cup \{i,j\}$ then it is either separated from $i$ by $A$ or from $j$ by $A$ or both otherwise there would be a path from $i$ to $j$ not having a vertex in $A$. We may therefore partition the remaining vertices into subsets $B$ and $C$ as in Figure 2.8 where:

$B = \{\ l\ ;\ l$ is not connected to $A \cup \{i,j\}$ or $l$ is separated from $j$ by $A\ \}$
$C = \{\ l\ ;\ l$ is not in $B$ and $l$ is separated from $i$ by $A\ \}$

Now let us map the independence graph of $X$ into the blocked graph of Figure 2.9 with vertices $\{i,j,A,B,C\}$. We draw an edge between any pair if an edge can exist between the elements of that pair in the original independence graph. No

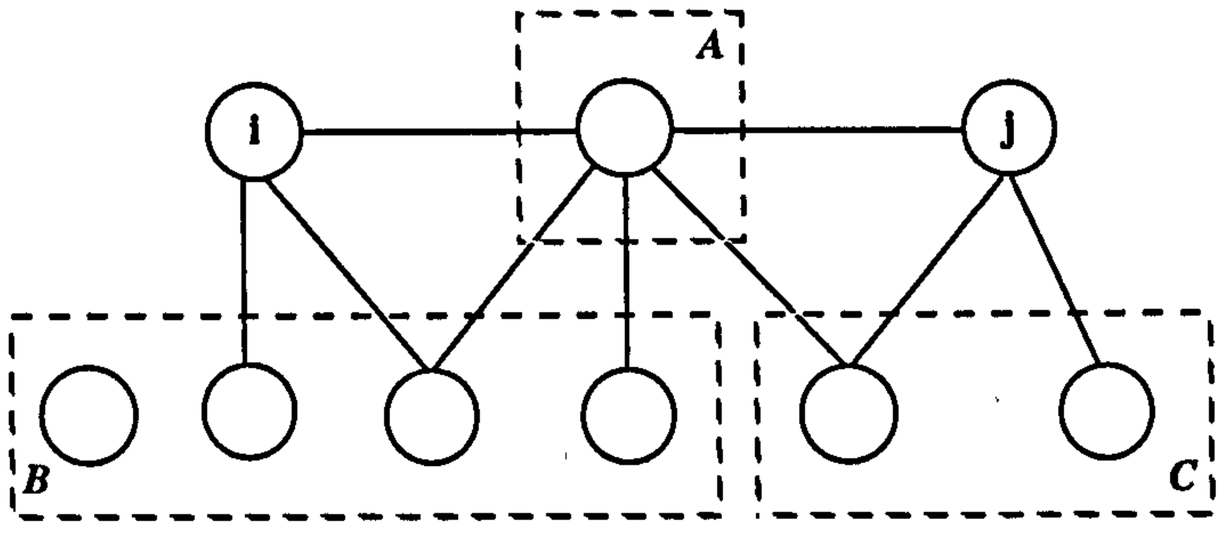



Figure 2.8: An independence graph to illustrate Lemma 2.

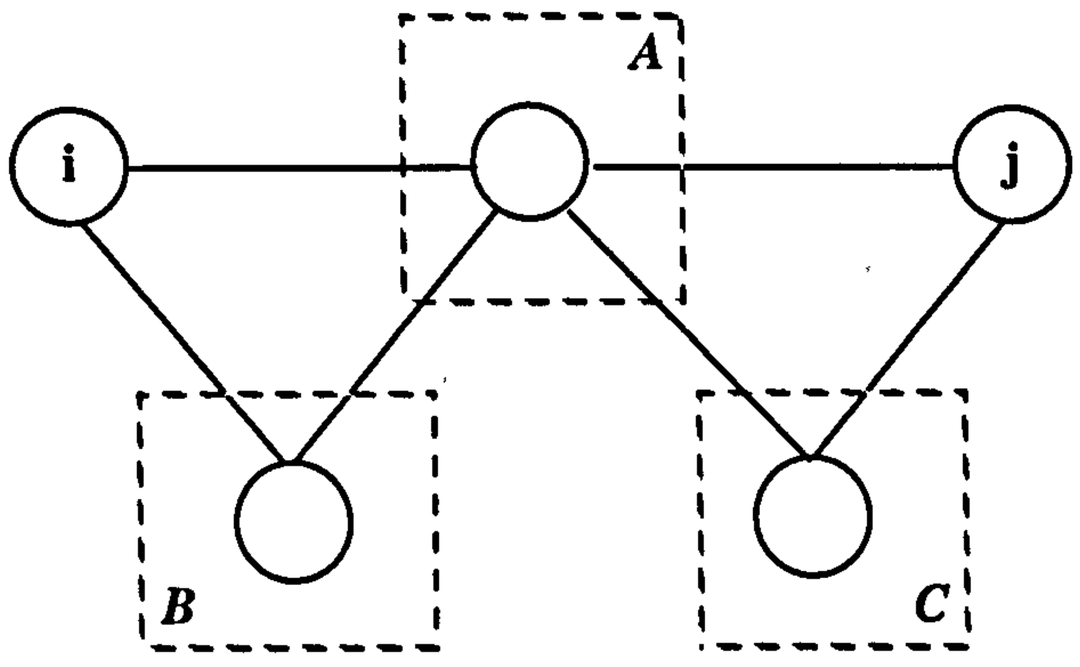



Figure 2.9: A blocked graph to illustrate Lemma 2.

edge may exist between $i$ and $j$ by hypothesis, and by construction no edges may exist between $i$ and $C$, $j$ and $B$, or $B$ and $C$.

Consider two vertices $p$ and $q$ in $C$ for $p \neq q$, and one vertex $r \in B$. The construction of the independence graph implies that:

$$
\begin{aligned}
&i \perp\!\!\!\perp j \mid K \backslash \{i, j\} && p \perp\!\!\!\perp r \mid K \backslash \{p, r\} \\
&i \perp\!\!\!\perp p \mid K \backslash \{i, p\} && q \perp\!\!\!\perp r \mid K \backslash \{q, r\} \\
&i \perp\!\!\!\perp q \mid K \backslash \{i, q\}
\end{aligned}
$$

So by the application of block independence and reduction we have:

$$
\left.\begin{aligned} &i \perp\!\!\!\perp j \mid K \backslash \{i, j\} \\ &i \perp\!\!\!\perp p \mid K \backslash \{i, p\} \end{aligned}\right\} \Rightarrow i \perp\!\!\!\perp \{j, p\} \mid K \backslash \{i, j, p\} \Rightarrow i \perp\!\!\!\perp j \mid K \backslash \{i, j, p\}
$$

$$
\left.\begin{aligned} &i \perp\!\!\!\perp p \mid K \backslash \{i, p\} \\ &i \perp\!\!\!\perp q \mid K \backslash \{i, q\} \end{aligned}\right\} \Rightarrow i \perp\!\!\!\perp \{p, q\} \mid K \backslash \{i, p, q\} \Rightarrow i \perp\!\!\!\perp q \mid K \backslash \{i, p, q\}
$$

$$\left.\begin{array}{l} j \perp\!\!\!\perp r \mid K\backslash\{j,r\} \\ p \perp\!\!\!\perp r \mid K\backslash\{p,r\} \end{array}\right\} \Rightarrow r \perp\!\!\!\perp \{j,p\} \mid K\backslash\{j,p,r\} \Rightarrow r \perp\!\!\!\perp j \mid K\backslash\{j,p,r\}$$

$$\left.\begin{array}{l} p \perp\!\!\!\perp r \mid K\backslash\{p,r\} \\ q \perp\!\!\!\perp r \mid K\backslash\{q,r\} \end{array}\right\} \Rightarrow r \perp\!\!\!\perp \{p,q\} \mid K\backslash\{p,q,r\} \Rightarrow r \perp\!\!\!\perp q \mid K\backslash\{p,q,r\}$$

Thus in the independence graph of $K\backslash\{p\}$ there is no edge induced between $i$ and $j$, nor between $i$ and any element in $C\backslash\{p\}$, nor between $j$ and any element in $B$, nor between any element in $B$ and any element in $C\backslash\{p\}$. Thus the structure is unchanged on removing the vertex $p$.

Repeating this argument with further vertices $p$ in $C$ and then, equivalently, with vertices $p$ in $B$ we may remove all the vertices from $B$ and $C$ to leave $i \perp\!\!\!\perp j \mid A$.

□

Since the construction of Lemma 2 works equally well with random vectors (i.e. sets of vertices) rather than single variables it immediately generalises for $X_B$ and $X_C$ replacing $X_i$ and $X_j$. We have thus proved:

**Theorem 1 The Separation Theorem:** *If $X_A$, $X_B$ and $X_C$ are vectors containing disjoint subsets of variables from $X$, and if, in the independence graph of $X$, each vertex in $B$ is separated from each vertex in $C$ by the subset $A$ then:*

$$X_B \perp\!\!\!\perp X_C \mid X_A$$

*assuming that $X$ has a positive joint density.*


Figure 2.10: An independence graph to illustrate the separation theorem.

Redundancies may exist in the conditioning variables in conditional independence relationships. A conditional independence between a pair of variables is said to be *minimal* if it is not possible to apply the separation theorem to eliminate any variable from the conditioning set. The separating set is then termed a *minimal separating subset*. Consider the example of Whittaker (1990). The conditional independence statements in the graph of Figure 2.10 are as follows:

$$1 \perp\!\!\!\perp 3 \mid \{2,4\},\ 1 \perp\!\!\!\perp 4 \mid \{2,3\}, \text{ and } 2 \perp\!\!\!\perp 4 \mid \{1,3\}$$

But, the minimal independences by application of the separation theorem are simply:

$$1 \perp\!\!\!\perp 3 \mid 2,\ 1 \perp\!\!\!\perp 4 \mid 2,\ 1 \perp\!\!\!\perp 4 \mid 3, \text{ and } 2 \perp\!\!\!\perp 4 \mid 3$$

## 2.6 Markov Properties

In this section we introduce three different conditional independence properties which an independence graphs might possess. These are the *local*, *pairwise* and *global Markov* properties (see Whittaker, 1990; Lauritzen *et al.*, 1990).

**Local Markov Property** The *local Markov property* expresses the independence statements in a graph in terms of a specified vertex and its nearest neighbours in the boundary set. A random vector with graph $\mathcal{G}$ is said to possess the local Markov property, if, for every vertex $i$ with boundary $A = bd(i)$, and $B$ the set of remaining vertices, then:

$$X_i \perp\!\!\!\perp X_B \mid X_A \text{ where } B = K \backslash (\{i\} \cup A)$$

This may be written $i \perp\!\!\!\perp rest \mid boundary$. We thus have $k$ independence statements about the joint distribution of all $k$ variables since $\{i\} \cup A \cup B = K$. This list of independences may, however, contain some redundancies. Consider the example of Whittaker (1990). We have an undirected graph with five vertices (see Figure 2.11).

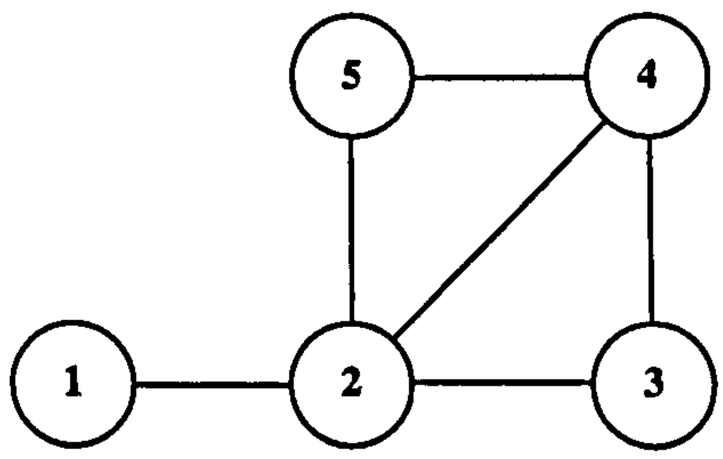



Figure 2.11: A graph to illustrate the local, pairwise and global Markov properties.

By the local Markov property we may express the following independences:

$$1 \perp\!\!\!\perp \{3,4,5\} \mid 2 \quad \text{(i)} \qquad 3 \perp\!\!\!\perp \{1,5\} \mid \{2,4\} \quad \text{(ii)}$$
$$4 \perp\!\!\!\perp 1 \mid \{2,3,5\} \quad \text{(iii)} \qquad 5 \perp\!\!\!\perp \{1,3\} \mid \{2,4\} \quad \text{(iv)}$$

But there is redundancy here since the independence between $X_3$ and $X_5$ given $\{X_2, X_4\}$ occurs in both (ii) and (iv). Similarly (iv) implies that $X_1 \perp\!\!\!\perp X_5 \mid (X_2, X_4)$ but (i) implies that $X_1 \perp\!\!\!\perp X_5 \mid X_2$. So we have a further redundancy.

**Pairwise Markov Property** The *pairwise Markov property* applies to pairs of non-adjacent vertices $i$, $j$ in the vertex set $K$. A graph $\mathcal{G}$ is said to possess the pairwise Markov property if for all such pairs $i$, $j$:

$$X_i \perp\!\!\!\perp X_j \mid X_A \text{ where } A = K \backslash \{i, j\}$$

For the graph of Figure 2.11 we have the following pairwise independences:

$$1 \perp\!\!\!\perp 3 \mid \{2,4,5\} \qquad 1 \perp\!\!\!\perp 4 \mid \{2,3,5\}$$
$$1 \perp\!\!\!\perp 5 \mid \{2,3,4\} \qquad 3 \perp\!\!\!\perp 5 \mid \{1,2,4\}$$

**Global Markov Property** The *global Markov property* holds for a graph $\mathcal{G}$ if for all disjoint subsets $A$, $B$, $C$ of $K$, where $B$ and $C$ are separated by $A$ in the graph, $X_B$ and $X_C$ are independent given $X_A$ alone:

$$X_B \perp\!\!\!\perp X_C \mid X_A$$

For the graph of Figure 2.11 we have the following global independences:

$$1 \perp\!\!\!\perp 3 \mid 2 \qquad 1 \perp\!\!\!\perp 4 \mid 2$$
$$1 \perp\!\!\!\perp 5 \mid 3 \qquad 3 \perp\!\!\!\perp 5 \mid \{2,4\}$$

**Theorem 2** *The three Markov properties: pairwise Markov, local Markov and global Markov, are equivalent.*

**Proof** Since the boundary set of a vertex $i$ is always a separating subset the global Markov property implies the local Markov property.

The local Markov property implies the pairwise Markov property because if our graph with vertices $K = \{1, 2, \ldots, k\}$ satisfies the local Markov property then for every vertex $i$ with boundary set $A = bd(i)$

$$X_i \perp\!\!\!\perp X_B \mid X_A \text{ where } B = K \backslash (\{i\} \cup A)$$

Selecting any vertex $j$ that is not adjacent to $i$ then $j \in B$. Letting $C = B \backslash \{j\} = K \backslash (\{i, j\} \cup A)$ then:

$$X_i \perp\!\!\!\perp (X_j, X_C) \mid X_A$$

But, assuming the joint density of $X$ is positive, by block independence this is equivalent to:

$$X_i \perp\!\!\!\perp X_j \mid (X_A, X_C) \text{ and } X_i \perp\!\!\!\perp X_C \mid (X_A, X_j)$$

However, $A \cup C = K \backslash \{i, j\}$ so the first independence statement is just the pairwise independence property.

Finally, the separation theorem asserts that the pairwise conditional independence property implies the global Markov property. Hence the three properties are equivalent.

□

## 2.7 Directed Acyclic Independence Graphs

We have so far considered only independence graphs with undirected edges between vertices. Such graphs may only be used to express symmetric associations between pairs of variables. Consider the case for example where we are measuring the interaction of two hormones in the body. It may be that these regulate each other and the level of one thus controls the level of the other and vice versa. In this case it would be perfectly justifiable to expect a symmetric association to exist between them. There are many examples, however, where we may not contain ourselves to this symmetry, where the roles played by the variables correspond to a notion of *causality* in that if $X$ causes $Y$ then $Y$ cannot cause $X$. Consider, for example, the presence of lung cancer in a patient and whether he smokes or not. It is reasonable to expect smoking to cause lung cancer but not for lung cancer to cause smoking. We therefore require a method of incorporating such asymmetries and this is provided by the use of a directed independence graph. A graph $\mathcal{G} = (K, E)$ has a directed edge, indicating a causal relationship, from $a$ to $b$ if $(a, b) \in E$ and $(b, a) \notin E$. The directed graphs with edge sets $\{1, 2\}$ and $\{2, 1\}$ are shown in Figures 2.12 and 2.13 respectively.

We are unable to include directed cycles in our graphs since there is, in general, no suitable joint probability to model such a relationship and we thus confine


Figure 2.12: The directed acyclic graph with edge set $\{1,2\}$.

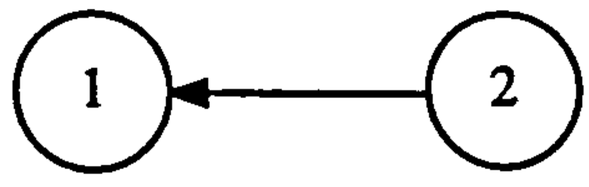



Figure 2.13: The directed acyclic graph with edge set $\{2,1\}$.

ourselves to *directed acyclic graphs* (*DAGs*). The removal of directed cycles is equivalent to supposing that the vertices of the graph may be *well-numbered*. The vertices of a graph $\mathcal{G} = (K, E)$ are said to be *well-numbered* if for every pair of distinct vertices $\{i, j\} \in K$ $i < j$ if and only if $i$ is an ancestor of $j$.

**Lemma 3** *A directed graph is acyclic if and only if the vertices can be well numbered.*

**Proof** If a graph can be well-numbered then it must be acyclic, since if a directed cycle exists then we may find a vertex $i$ that leads to itself so we may derive $i < i$ which is clearly not true.

For the converse we may well number an acyclic graph as follows: A DAG must have at least one *root node* (that is a node with no parents) and one *leaf node* (that is one with no children). Choose a root node and label it 1. Delete it from the graph $\mathcal{G}$ to obtain the DAG $\mathcal{G}'$. We may then repeat our argument by choosing a root node of $\mathcal{G}'$, labelling it 2, and deleting it. Continuing in this way an ordering which is well-numbered may be deduced from an unnumbered graph. Alternatively we could have chosen leaf nodes and labelled them $k$ , $k-1$, ..., 1 deleting them in turn.

□

We will now introduce the precedence operator $\prec$. If $i$ and $j$ are two distinct vertices in a DAG $\mathcal{G}$ then $i \prec j$ indicates that $i$ is an ancestor of $j$. We term the DAG *completely ordered* since it may be well-numbered. If $\mathcal{G}$ is well-numbered then $i < j$ also.

**Definition 4** *The directed independence graph (recursive graph) of a vector of random variables* $X = (X_1, X_2, \ldots, X_k)$ *is the directed graph* $\mathcal{G} = (K, E)$ *where* $K = \{1, 2, \ldots, k\}$, $K(j) = \{1, 2, \ldots, j\}$ *and the edge* $(i, j)$, *with* $i \to j$, *is not in the edge set* $E$ *if and only if* $j \perp\!\!\!\perp i \mid K(j) \backslash \{i, j\}$.

This is the same definition as that which we used for undirected independence graphs except we have modified our conditioning set to comprise of just the 'past' as opposed to the 'past' and 'future' we previously termed the 'rest'. This sequential conditioning allows us to form the joint distribution of $X$ by use of the *recursive factorisation identity* (or *chain rule*):

$$f_{1,2,\ldots,k} = f_{k|K(k)\setminus\{k\}} f_{k-1|K(k-1)\setminus\{k-1\}} \cdots f_{2|1} f_1$$

### 2.7.1 Moralisation

We require a method of connecting our undirected independence graph theory to that of our directed independence graph theory. This will allow us to simplify our problems by enabling us to work with undirected graphs alone. We therefore introduce the concept of a moral graph.

**Definition 5** *The* moral graph *associated with the directed graph $\mathcal{G} = (K, E)$ is the undirected graph $\mathcal{G}^m = (K, E^m)$ on the same vertex set and with an edge set obtained by including all the edges in $E$ together with all the edges formed by marrying parents in $\mathcal{G}$.*


Figure 2.14: A directed acyclic graph and its corresponding moral graph.

Figure 2.14 shows a directed acyclic graph and its corresponding moral graph. The importance of moral graphs is expressed in the following theorem:

**Theorem 3** *The directed independence graph $\mathcal{G}$ possesses the Markov properties of its associated moral graph $\mathcal{G}^m$.*

**Proof** Consider the recursive factorisation identity:

$$\begin{aligned} f_{1,2,\ldots,k} &= f_{k|K(k)\setminus\{k\}} f_{k-1|K(k-1)\setminus\{k-1\}} \cdots f_{2|1} f_1 \\ &= f_1 \prod_{j=2}^{k} f_{j|K(j)\setminus\{j\}} \\ &= f_1 \prod_{j=2}^{k} f_{j|pa(j)} \\ &= g_1 \prod_{j=2}^{k} g_{j\cup pa(j)} \end{aligned}$$

For appropriate functions $g$. We thus have an expansion of the joint density in terms of functions $g_A$, which are functions of $x_A$ for $A = \{1\}, \{2\} \cup pa(2), \ldots, \{k\} \cup pa(k)$. We may now apply the factorisation criterion (see Proposition 3) to deduce all pairwise conditional independence statements of the form $i \perp\!\!\!\perp j \mid rest$, i.e.:

$$\begin{aligned} f_K &= \cdots g_{i \cup pa(i)} \cdots g_{j \cup pa(j)} \cdots \\ &\Leftrightarrow \quad i \perp\!\!\!\perp j \mid (pa(i) \cup pa(j)) \backslash \{i, j\} \text{ for all } i \neq j \\ &\Rightarrow \quad i \perp\!\!\!\perp j \mid K \backslash \{i, j\} \end{aligned}$$

So the edges of the undirected independence graph for $f_K$ are characterised as edges between a vertex $j$ and each of its parents, and edges between the members of each pair of parents of $j$. That is, the edge set of the moral graph, $\mathcal{G}^m$.

□

So as a consequence of Theorem 3 we may concern ourselves with the moral graph $\mathcal{G}^m$ of our directed independence graph $\mathcal{G}$ and thus need only work in the undirected case.

## 2.8 Chain Graphs

We have so far considered independence graphs which possess either undirected or directed edges. We now wish to extend our theory to cover graphs which contain a mixture of both undirected and directed edges. This will provide a generalisation of our independence graph theory which contains both of our special cases.

We assume that our vertex set will satisfy a *partial ordering*, as opposed to the complete ordering we imposed on directed graphs. We suppose that our vertex set $K$ may be partitioned into subsets $B_1, B_2, \ldots, B_m$, called *blocks*, and that these blocks are well-numbered and form a *chain*. Within a particular block we have no directed edges. Therefore, two elements from different blocks may only be joined by an arrow, and two elements from the same block may only be joined by a line. The *parents* of a vertex $i$ in a block $B_r$ are drawn from the 'past', $B_1 \cup B_2 \cup \ldots \cup B_{r-1}$, and are joined to $i$ by directed edges. The elements in $B_1$ are the *potential causes* of the elements in $B_2$, the elements in $B_1 \cup B_2$ are the potential causes of the elements in $B_3$, and so on. We may also use the operators $\prec$ (as defined above) and $\preceq$ to define our partial ordering. If $i \preceq j$ then $i$ and $j$ are in the same block. The induced partial order on the vertices of $K$ is such that $i \prec j$ whenever $i \in B_r$, $j \in B_s$ and $r < s$; and $i \preceq j$ whenever $i, j \in B_r$.

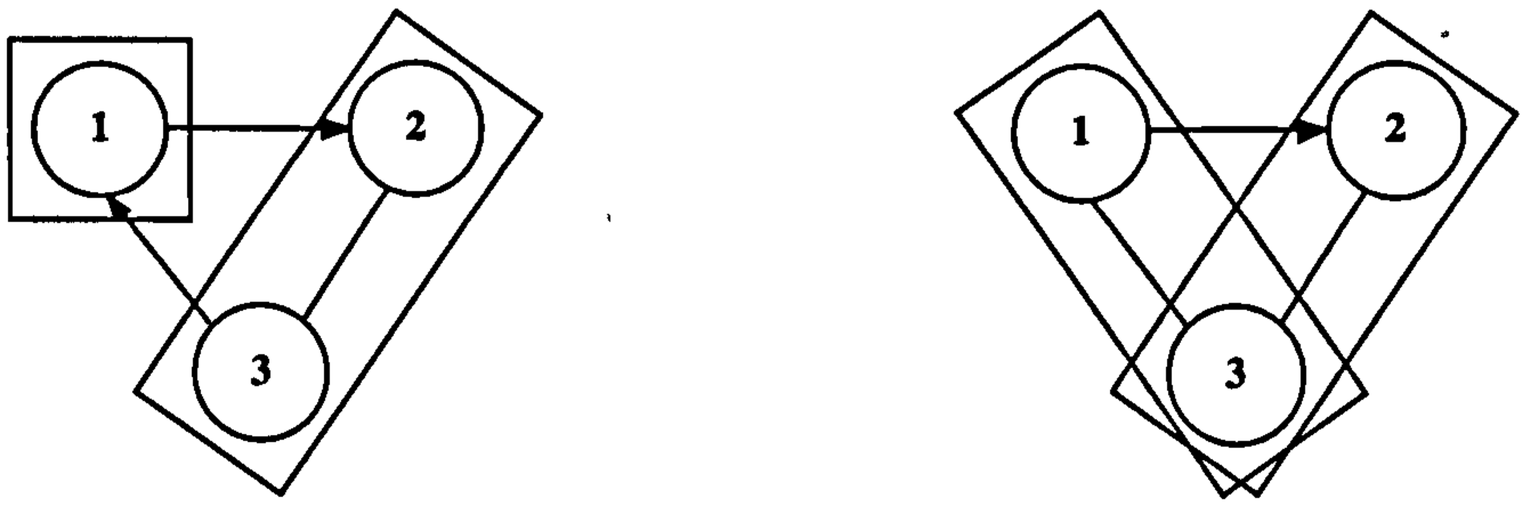



Figure 2.15: Two graphs which may not be represented as chain graphs.

We have, by construction, included the directed independence graph case. That is where blocks have only one element contained within them. The undirected case has also been dealt with, since this would consist of a chain graph with only one block. We have also excluded graphs with cycles containing at least one directed edge (see Frydenberg, 1989). These do not admit a reasonable factorisation of the joint density function and would require at least one vertex to exist in more than one block. In Figure 2.15 we consider two disallowed configurations of the vertex set $K = \{1, 2, 3\}$. In the first graph we cannot form a complete ordering of our blocks since we have a cycle, so this type of graph is not allowed. In the second graph vertex 3 would need to be contained in both blocks which again is not permitted by our construction.

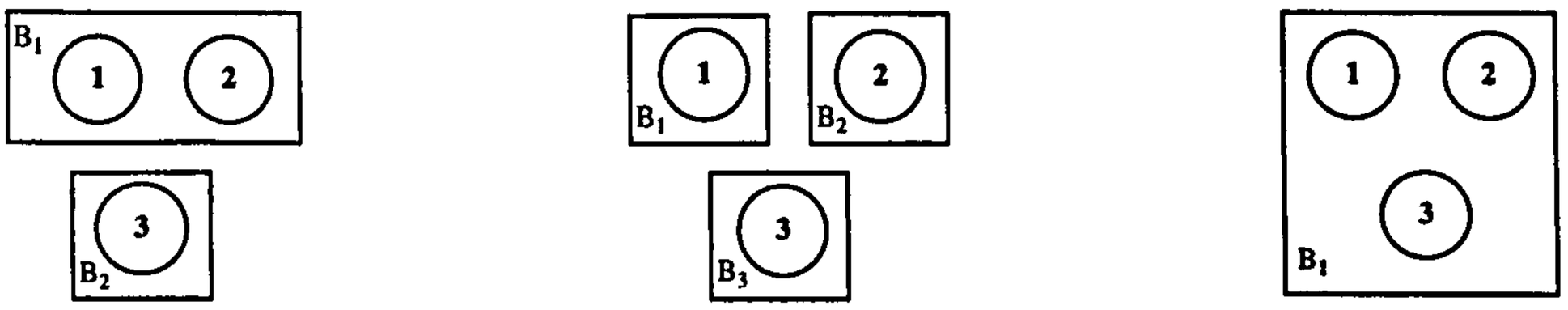
We may now define the chain independence graph from the pairwise conditional independences taking the conditioning set for each statement as $K(i)$ - the *concurrent past*. This consists of all the 'past' and 'present' variables defined with respect to a vertex $i$ as $K(i) = B_1 \cup B_2 \cup \ldots \cup B_{r(i)}$ where $r(i)$ is the index of the block containing $i$.

**Definition 6** *The* **chain independence graph** *of a random vector* $X = (X_1, X_2, \ldots, X_k)$ *is the graph* $\mathcal{G} = (K, E)$, *where* $K = \{1, 2, \ldots, k\}$, $K(i) = \bigcup_{l \leq r(i)} B_l$, *and the edge* $(i, j)$, *with* $i \preceq j$, *is not in the edge set* $E$ *if and only if* $j \perp\!\!\!\perp i \mid K(j) \backslash \{i, j\}$. *If this condition fails and* $i \prec j$, *then the edge is directed so* $(i, j) \in E$ *and* $(j, i) \notin E$; *otherwise it is undirected and both* $(i, j) \in E$ *and* $(j, i) \in E$.

Such graphs are also called *block recursive* by Lauritzen & Wermuth (1984, 1989). An alternative approach to the definition for a chain graph is to allow a mixture of directed and undirected edges subject to the constraint that no cycle containing one or more directed edge is permitted. We then have the following definition:

**Definition 7** *A* **chain graph** *is a graph whose vertex set $K$ can be partitioned into a series of $m$ numbered subsets $B_1, B_2, \ldots, B_m$. These subsets form a* dependence chain $K = B_1 \cup B_2 \cup \ldots \cup B_m$ *such that all edges between vertices in the same subset are undirected and all edges between different subsets are directed, pointing from the set with lower number to the one with higher number.*

Such a block structure is not unique as may be seen in Figure 2.16.




$3 \perp\!\!\!\perp 2 \mid 1$ $\quad$ $3 \perp\!\!\!\perp 2 \mid 1$ $\quad$ $1 \perp\!\!\!\perp 2 \mid 3$

$3 \perp\!\!\!\perp 1 \mid 2$ $\quad$ $3 \perp\!\!\!\perp 1 \mid 2$ $\quad$ $1 \perp\!\!\!\perp 3 \mid 2$

$1 \perp\!\!\!\perp 2$ $\quad$ $2 \perp\!\!\!\perp 1$ $\quad$ $2 \perp\!\!\!\perp 3 \mid 1$

Figure 2.16: Three alternative block structures with common vertex and edge sets.

We also notice that the initial pairwise independences are different. It has been shown by Frydenberg (1989) that all consistent block representations are probabilistically equivalent. He also shows that it is the connectivity components (see earlier) that form the finest possible partitioning of the graphs. This means that the connectivity components are the *chain components* of the graph, i.e. the subsets $B_1, B_2, \ldots, B_m$. Chain components can be most easily found by removing all the arrows from the graph before taking the connectivity components.

### 2.8.1 Markov Properties of Chain Graphs

Recall from earlier that in order to form the undirected graph corresponding to a directed graph we first married the parents of each vertex in the graph. This condition is extended for chain graphs by including in the set of parents all the parents of a connected subset of children.

**Definition 8** *The* **moral graph** $\mathcal{G}^m$ *of a chain graph $\mathcal{G}$ is defined to be the undirected graph with the same vertex set as $\mathcal{G}$ but with $i$ and $j$ adjacent in $\mathcal{G}^m$ if and only if either $i \rightarrow j$ or $j \rightarrow i$ or if there exists $p$, $q$, connected, in the same chain component such that $i \rightarrow p$ and $j \rightarrow q$.*

Thus in the chain independence graph of Figure 2.17 we must marry vertices 1 and 2, since they are the parents of the connected children 3 and 5, to form the associated moral graph.


Figure 2.17: A chain independence graph and its corresponding moral graph.

In the chain independence graph of Figure 2.18 in order to form the associated moralised graph we must marry vertices 3 and 4 since they are parents of vertex 5, but vertices 1 and 2 are not married since vertices 3 and 4 are not connected. Recall our earlier example of Figure 2.5 here we add a link between vertices 1 and 2 to form the moral graph of Figure 2.19.


Figure 2.18: A chain independence graph and its corresponding moral graph.

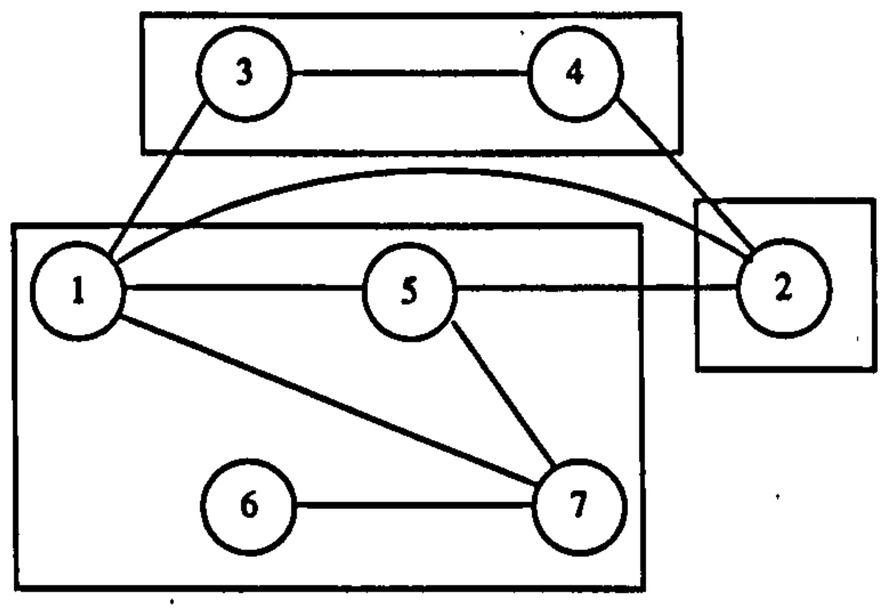



Figure 2.19: The moral graph corresponding to the graph in Figure 2.5.

We may define the *recursive factorisation identity* (or *chain rule*) in terms of the blocks of our chain graphs thus:

$$f_{1,2,\ldots,k} = f_K = f_{B_1} \prod_{r=2}^{m} f_{B_r|B_1 \cup B_2 \cup \ldots \cup B_{r-1}}$$

**Theorem 4** *The chain independence graph $\mathcal{G}$ possesses the Markov properties of its associated moral graph $\mathcal{G}^m$.*

**Proof** Let $\mathcal{G} = (K, E)$ where $K = \{1, 2, \ldots, k\}$ and let $B_1, B_2, \ldots, B_m$ be the blocks of $\mathcal{G}$. Now consider two arbitrary vertices $i$ and $j$ of $K$ and let us assume either $i \prec j$ or $i \preceq j$. Then, by construction, either $i \in B_r$, and $j \in B_s$, with $r < s$, or both $i, j \in B_r$. Where an edge exists between $i$ and $j$ it will be an arrow in the first case and a line in the second. We will let $\mathcal{G}^m = (K, E^m)$ denote the moral graph of $\mathcal{G}$ and this moral graph will be undirected.

Let us consider the properties that $\mathcal{G}^m$ will need to possess in order that all its Markov properties apply to $\mathcal{G}$. Firstly all the edges in $\mathcal{G}$ will need to be preserved in $\mathcal{G}^m$. This is because if the structure of $f_{B_r|B_1 \cup B_2 \cup \ldots \cup B_{r-1}}$ does not permit a factorisation of $j$ from $i$ then, by the recursive factorisation identity, such a factorisation is also prohibited for $f_K$. Therefore if $j$ and $i$ are joined by an edge in $\mathcal{G}$ then they must be connected in $\mathcal{G}^m$.

Now suppose we have connected the pairs of vertices joined by edges in $\mathcal{G}$. Which other lines do we need to add to $\mathcal{G}^m$? Suppose that currently the edge between $i$ and $j$ is missing in $\mathcal{G}^m$ we need only add it if there is some block $r$ such that $f_{B_r|B_1 \cup B_2 \cup \ldots \cup B_{r-1}}$ prevents the factorisation of $i$ and $j$. Let $i$ be the parent of some vertex $p \in B_r$ and $j$ be the parent of some vertex $q \in B_r$. Then if $p$ and $q$ are not connected we may partition $B_r$ into two sets, $C_1$ containing $p$ and $C_2$ containing $q$, such that:

$$f_{B_r|B_1 \cup B_2 \cup \ldots \cup B_{r-1}} = f_{C_1|B_1 \cup B_2 \cup \ldots \cup B_{r-1}} f_{C_2|B_1 \cup B_2 \cup \ldots \cup B_{r-1}}$$

Since $i$ is a parent of $p$ and not of $q$ then $f_{C_1|B_1 \cup B_2 \cup \ldots \cup B_{r-1}}$ is a function of $i$ but not of $j$. Similarly, since $j$ is a parent of $q$ and not of $p$ then $f_{C_2|B_1 \cup B_2 \cup \ldots \cup B_{r-1}}$ is a function of $j$ but not of $i$. So $f_{B_r|B_1 \cup B_2 \cup \ldots \cup B_{r-1}}$ factorises into the product of a function of $i$ and a function of $j$ hence $i$ and $j$ need not be connected.

If $p$ and $q$ are connected, however, we may not factorise $f_{B_r|B_1 \cup B_2 \cup \ldots \cup B_{r-1}}$ into the product of a function of $p$ and a function of $q$ and hence we cannot factorise it into the product of a function of $i$ and a function of $j$. An edge is therefore required between $i$ and $j$. The same will trivially apply if $p = q$.

But our moral graph was defined in Definition 8 with exactly this construction. Hence the proof.

□

Since all the Markov properties of the moral graph apply to the chain graph we need only concern ourselves with an undirected graph once moralisation has taken place.

Further details of graphical chain models may be found in Wermuth & Lauritzen (1989) and Whittaker (1990).

## 2.9 Marked Graphs

We have so far only concerned ourselves with the type of edges that join the vertices of our graphs and have not dealt with the types of variables that are associated with those vertices. We now introduce the concept of a *marked graph* where our vertex set is partitioned into two groups: one containing all the discrete (*qualitative*) variables, the other containing the continuous (*quantitative*) variables. We write this partition as follows:

$$K = \Delta \cup \Gamma$$

where $\Delta$ is the set of discrete variables and $\Gamma$ is the set of continuous variables. A special case of a marked graph is one which contains only one sort of variable (either all discrete, or all continuous), this we term a *pure graph*. We represent the different types of variables with different symbols. Discrete vertices are represented by dots (filled circles) and continuous vertices are represented by circles. Figure 2.20 shows a mixed chain graph. Our notation is equally valid in directed and undirected graphs also.

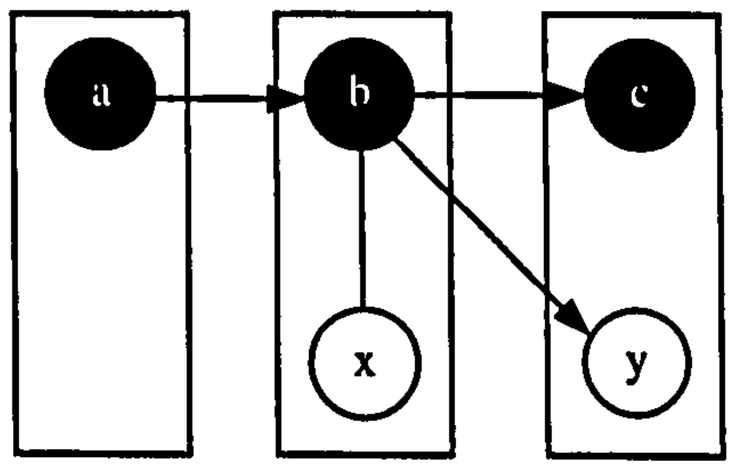



Figure 2.20: A marked graph.

In Figure 2.20 vertices $a$ , $b$ and $c$ represent discrete variables, while $x$ and $y$ represent continuous variables. The chain (connectivity) components $\{a\}$, $\{b, x\}$ , and $\{c, y\}$ are also shown.

The partitioning of variables into discrete or continuous type has no effect on our concepts of conditional independence as previously defined so no extensions are required to our existing theory.

## 2.10 Decomposable Models

### 2.10.1 Weak Decompositions and Weakly Decomposable Graphs

First let us consider undirected graphs whose vertex set is either marked or pure. Lauritzen (1989) defines a *weak decomposition* to be as follows:

**Definition 9** *A triple* $(A, B, C)$ *of disjoint subsets of the vertex set* $K$ *of an undirected graph* $\mathcal{G}$ *is said to form a* **weak decomposition** *of* $\mathcal{G}$ *if* $K = A \cup B \cup C$ *and the following conditions both hold:*

*i)* $C$ *separates* $A$ *from* $B$.
*ii)* $C$ *is a complete subset of* $K$.

Note that we allow some of the sets in the triple $(A, B, C)$ to be empty. If both $A$ and $B$ are non-empty then the weak decomposition is termed a *proper weak decomposition*. If $\mathcal{G}$ forms a weak decomposition then it is said to be *reducible*. Otherwise it is *irreducible*. In Figure 2.21 the triple $(A, B, C)$ forms a weak decomposition of $\mathcal{G}$. $\mathcal{G}$ weakly decomposes into the components $\mathcal{G}_{A \cup C}$ and $\mathcal{G}_{B \cup C}$ since $C$ is a complete subset of $K$ and separates $A$ from $B$. Also, since both $A$ and $B$ are non-empty this weak decomposition is proper. In Figure 2.22 the triple $(A, B, C)$ is not a weak decomposition since the separating set $C$ is not a complete subset of $K$.

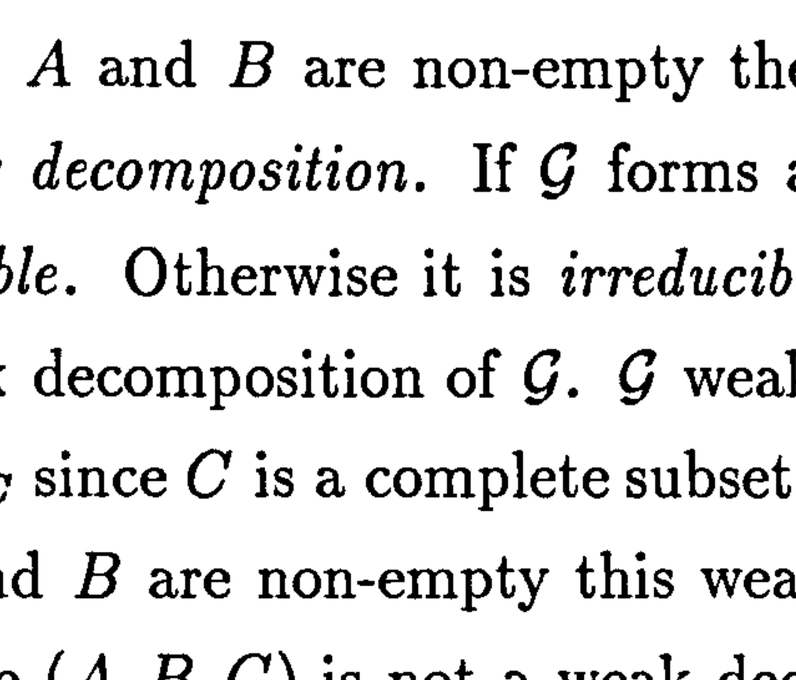



Figure 2.21: A graph which forms a weak decomposition $\{A, B, C\}$.

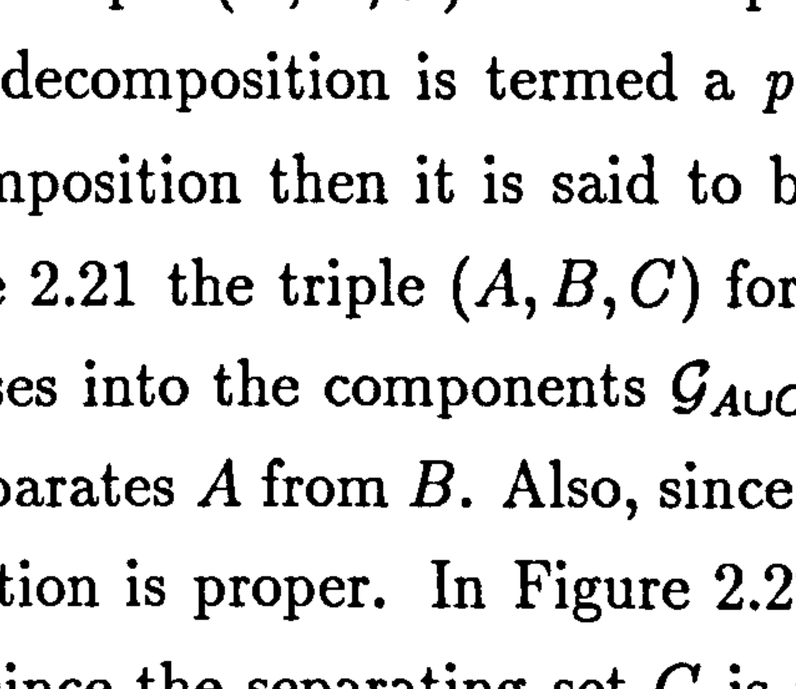



Figure 2.22: A graph which does not form a weak decomposition $\{A, B, C\}$.

Related to the concept of a weak decomposition are *weakly decomposable graphs* and *triangulated graphs*.

**Definition 10** *An undirected graph is said to be a* **weakly decomposable graph** *if it is complete, or if there exists a proper weak decomposition* $(A, B, C)$ *into weakly decomposable subgraphs* $\mathcal{G}_{A\cup C}$ *and* $\mathcal{G}_{B\cup C}$.

or, equivalently:

**Definition 11** *A* $k-$*dimensional random vector* $X$*, or its density function, is* **weakly decomposable** *if and only if there exists a sequence of decompositions to complete irreducible components.*

Pearl (1988) defines a *triangulated graph* (or *chordal graph*) as follows:

**Definition 12** *An undirected graph* $\mathcal{G} = (K, E)$ *is said to be* **triangulated** *or* **chordal** *if every cycle of length four or more has a chord, i.e. an edge joining two nonconsecutive vertices.*

The connection between weakly decomposable graphs and triangulated graphs is elucidated in the following theorem:

**Theorem 5 The Triangulation Theorem** *The random vector* $X$ *(or its density function) is weakly decomposable if and only if its independence graph* $\mathcal{G}$ *is triangulated.*

**Proof** We shall assume that $\mathcal{G}$ is connected, or else we may apply the proof to each of the connected subgraphs of $\mathcal{G}$ separately.

Assume that $\mathcal{G}$ is not triangulated then there exists a chordless $n-$cycle of $\mathcal{G}$ such that $n > 3$. Let $u$ and $v$ be two non-adjacent vertices in this $n-$cycle and suppose the other distinct vertices are $Q = \{q_1, q_2, \ldots, q_m\}$ and $R = \{r_1, r_2, \ldots, r_p\}$ (see Figure 2.23). Consider weakly decomposing any graph containing the $n-$cycle into two subgraphs $A$ and $B$ where $u \in A$ and $v \in B$. For the conditional independence property $A \perp\!\!\!\perp B \mid C$ to hold $A$ must include $q_1$ since it is adjacent to $u$ and on a path from $u$ to $v$. Similarly, $B$ must include $q_m$. The same argument may be applied to $q_2$ and $q_{m-1}$ leading us to add these to $A$ and $B$ respectively. We may continue in this way until there is an overlap in the assignment of $Q$ to $A$ and $B$, i.e. $C \subseteq Q$. Now we may repeat this exercise on the other path from $u$ to $v$ which passes through $R$. The separating subset $C$ thus formed must include at

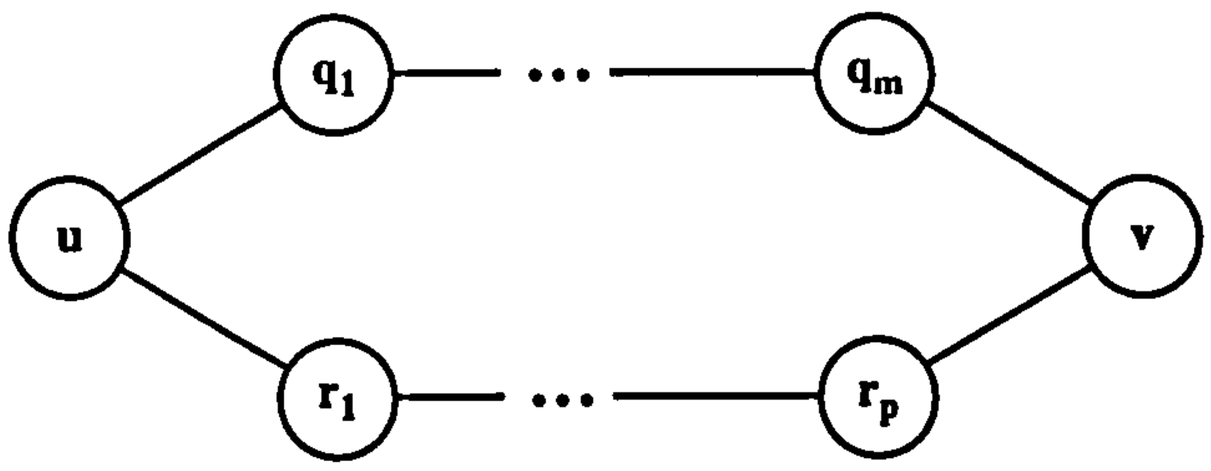



Figure 2.23: A graph to illustrate Theorem 5.

least one $q_i \in Q$ and $r_j \in R$. But since there is no chord in the $n-$cycle $q_i$ and $r_j$ are not connected and hence $C$ is not complete. Therefore no weak decomposition of a graph may destroy an $n-$cycle where $n > 3$. Trivially if $n = 3$ then since all three vertices in the 3-cycle are adjacent it too may not be broken by a weak decomposition. Thus if we weakly decompose $\mathcal{G}$ we will eventually end up with a subgraph containing the $n-$cycle which is irreducible but not complete. Hence if $\mathcal{G}$ is not triangulated then $\mathcal{G}$ is not decomposable.

Now suppose $\mathcal{G}$ is triangulated so there are no chordless $n-$cycles where $n > 3$ in $\mathcal{G}$. If $\mathcal{G}$ is complete then it is decomposable by definition. Otherwise consider two non-adjacent vertices $u$ and $v$ in $\mathcal{G}$. Let $Q = \{q_1, q_2, \ldots, q_m\}$ and $R = \{r_1, r_2, \ldots, r_p\}$ be two short paths from $u$ to $v$. Then if $\mathcal{G}$ is triangulated $q_1$ must be adjacent to $r_1$ otherwise a chordless cycle with at least four elements would exist. Let $C$ consist of the set of vertices adjacent to $u$ on short paths from $u$ to $v$ then $C$ will be complete since each of its vertices are adjacent. Let $A$ consist of $u$ and those vertices with paths to $u$ which do not intersect $C$. Finally let $B = K \backslash (A \cup C)$ so $A$, $B$ and $C$ are three disjoint subsets of $K = A \cup B \cup C$. Since $\mathcal{G}$ is connected any vertex in $A$ has a path to $u$, any vertex in $B$ has a path to $v$ and, by construction $u$ only has a path to $v$ via $C$. So $C$ separates $A$ from $B$ and hence $(A, B, C)$ forms a weak decomposition of $\mathcal{G}$ into subgraphs $\mathcal{G}_{A \cup C}$ and $\mathcal{G}_{B \cup C}$. Since $\mathcal{G}$ was triangulated and $n-$cycles where $n \geq 3$ are preserved under weak-decomposition then the two subgraphs are triangulated also. If both subgraphs are complete then $\mathcal{G}$ is weakly decomposable. Otherwise repeat the argument to any non-complete subgraph until the process terminates at which stage all subgraphs will be complete and irreducible. Hence $\mathcal{G}$ is decomposable if $\mathcal{G}$ is triangulated.

□

The following definition and theorem (proof omitted) come from Jensen (1996):

**Definition 13** *A node $A$ is* **eliminated** *by adding links such that all of its neighbours are pairwise linked and then removing $A$ together with its links.*

**Theorem 6** *A graph is triangulated if and only if all of its nodes can be eliminated one by one without adding any link.*

Theorem 6 provides a method for triangulation and a test to check whether a graph is triangulated. The nodes of a graph should be eliminated in some order (adding links if necessary) and the graph will then be triangulated. Kjærulff (1990) discusses the appropriateness of different elimination orderings.

### 2.10.2 Strong Decompositions and Strongly Decomposable Graphs

We will now extend the concept of a weak decomposition by considering the decomposition of a marked graph $\mathcal{G}$ introduced by Leimer (1989). Frydenberg & Lauritzen (1989) define a *strong decomposition* or *decomposition* as follows:

**Definition 14** *A triple $(A, B, C)$ of disjoint subsets of the vertex set $K$ of an undirected, marked graph $\mathcal{G}$ is said to form a* **decomposition** *or* **strong decomposition** *of $\mathcal{G}$ if $K = A \cup B \cup C$ and the following conditions all hold:*

*i)* $C$ *separates* $A$ *from* $B$.
*ii)* $C$ *is a complete subset of* $K$.
*iii)* $C \subseteq \Delta \vee B \subseteq \Gamma$.

We note that if $\mathcal{G}$ forms a decomposition then it also forms a weak decomposition since Definition 14 is the same as Definition 9 with the addition of condition (iii). In Figures 2.24 and 2.25 the triple $(A, B, C)$ decomposes $\mathcal{G}$ into the components $\mathcal{G}_{A\cup C}$ and $\mathcal{G}_{B\cup C}$. In both, $C$ separates $A$ from $B$ and $C$ is a complete subset of $K$. In Figure 2.24 the separating set $C$ contains only discrete variables, while in Figure 2.25 $A$ contains only continuous variables.

Related to the concept of a decomposition are *decomposable* or *strongly triangulated graphs*. Frydenberg & Lauritzen (1989) make the following, recursive, definition:

**Definition 15** *An undirected, marked graph is said to be* **decomposable** *or* **strongly triangulated** *if it is complete, or if there exists a proper decomposition $(A, B, C)$ into decomposable subgraphs $\mathcal{G}_{A\cup C}$ and $\mathcal{G}_{B\cup C}$.*

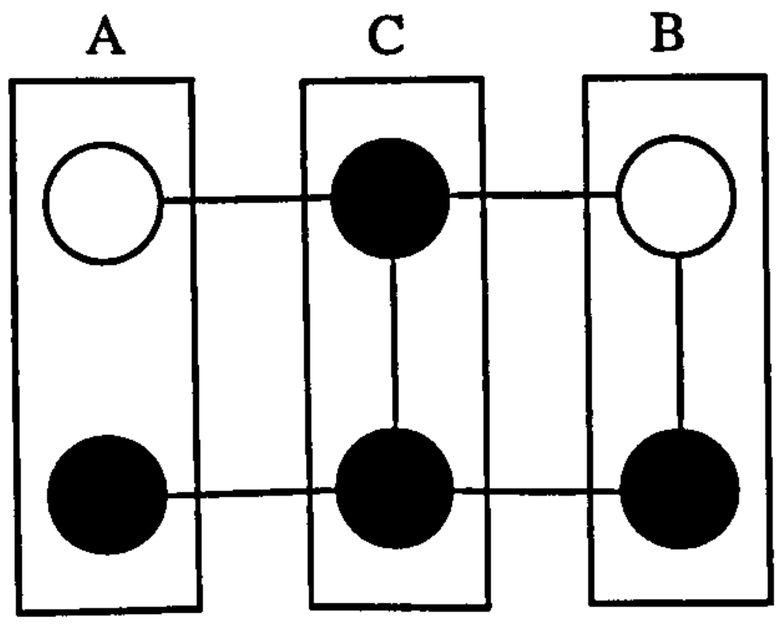



Figure 2.24: A graph forming a strong decomposition $\{A, B, C\}$.

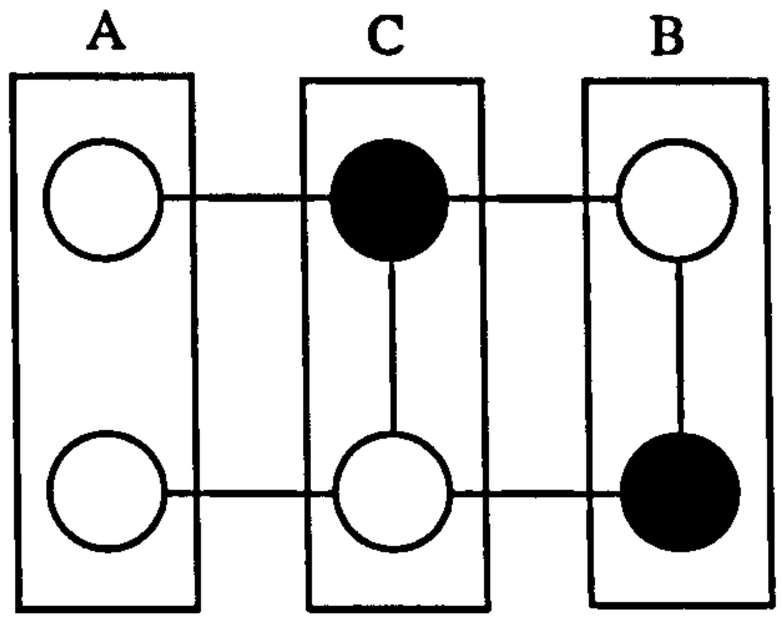



Figure 2.25: A second graph forming a strong decomposition $\{A, B, C\}$.

Again we note that since a decomposition is also a weak decomposition a decomposable graph is also weakly decomposable. Leimer (1989) proposes the following to be the case:

**Theorem 7** *An undirected marked graph is decomposable if and only if it is triangulated and does not contain any path* $(\delta_1 = \alpha_0, \ldots, \alpha_n = \delta_2)$ *between two non-adjacent discrete vertices passing through only continuous vertices.*

**Proof** By Theorem 5 a graph $\mathcal{G}$ is weakly decomposable if and only if it is triangulated. Hence $\mathcal{G}$ may only be decomposable if it is triangulated.

Let us suppose that $\mathcal{G}$ is triangulated and let it be connected or we may repeatedly apply our logic to connected subgraphs of $\mathcal{G}$. Consider two non-adjacent discrete vertices $\delta_1$ and $\delta_2$. Since $\delta_1$ and $\delta_2$ are non-adjacent, if $\mathcal{G}$ is to be decomposable then $\mathcal{G}$, or a graph decomposed from $\mathcal{G}$, will be required to form a decomposition $(A, B, C)$, with $\delta_1 \in A$ and $\delta_2 \in B$. Since $\mathcal{G}$ is triangulated $(A, B, C)$ will, at the very least, form a weak decomposition.

Now suppose that $\mathcal{G}$ contains a path $(\delta_1 = \alpha_0, \ldots, \alpha_n = \delta_2)$ between $\delta_1$ and $\delta_2$ which passes through only continuous vertices. Then there exists a short path between $\delta_1$ and $\delta_2$ which passes through only continuous vertices. Hence any $C$ which separates $A$ from $B$ must contain at least one continuous vertex. Thus $C \not\subseteq \Delta$ and since $\delta_1$ and $\delta_2$ are discrete $A \not\subseteq \Gamma$ and $B \not\subseteq \Gamma$. So $(A, B, C)$ cannot be a strong decomposition.

Suppose, however, that there is no path between two non-adjacent discrete vertices $\delta_1$ and $\delta_2$ which passes through only continuous vertices. Then any short path between two non-adjacent discrete vertices will pass through only discrete vertices. Thus any separating subset $C$ must have the property $C \subseteq \Delta$ and hence

$(A, B, C)$ is a strong decomposition. Let $C$ consist of the set of vertices adjacent to $\delta_1$ on short paths from $\delta_1$ to $\delta_2$, let $A$ consist of $\delta_1$ and those vertices with paths to $\delta_1$ which do not intersect $C$ and let $B = K \backslash (A \cup C)$. If we apply this procedure repeatedly to $\mathcal{G}$, and any decompositions of $\mathcal{G}$ thus formed, we will eventually obtain a set of subgraphs which are either complete and hence decomposable or have no non-adjacent discrete vertices. Let $\mathcal{G}' = (K', E')$ be a subgraph of this latter type with discrete vertices $\Delta'$.

If $\mathcal{G}'$ is pure then any decomposition of $\mathcal{G}'$ will be strong since $B \subseteq \Gamma$. If $\mathcal{G}'$ has either only one or two discrete vertices then either $A$ or $B$ or both will not contain any discrete vertices. Therefore either $A \subseteq \Gamma$ or $B \subseteq \Gamma$, or both so $\mathcal{G}'$ is decomposable. If $\mathcal{G}'$ has three or more discrete vertices then since 3-cycles are preserved under weak decompositions either $\Delta' \subseteq C$ and hence $B \subseteq \Gamma$; or $\Delta' \subseteq A$ and hence $B \subseteq \Gamma$; or $\Delta' \subseteq B$ and hence $A \subseteq \Gamma$. Thus $\mathcal{G}'$ forms a strong decomposition. Repeating this argument we may show that any $\mathcal{G}'$ is decomposable, and hence $\mathcal{G}$ is decomposable.

□

## 2.11 Junction Trees

We have shown how directed acyclic graphs and chain graphs may be moralised to form undirected graphs, and how these undirected graphs may then be triangulated (or strongly triangulated) to form weakly decomposable (or decomposable) graphs. In this section we will consider how we may form a higher level structure from these weakly decomposable graphs. First let use consider an example from Pearl (1988) which demonstrates a desirable property of a particular type of structure.

Suppose we have a distribution $P(x_1, \ldots, x_4)$ with a Markov network $\mathcal{G}$ in the form of a chain (see Figure 2.26).

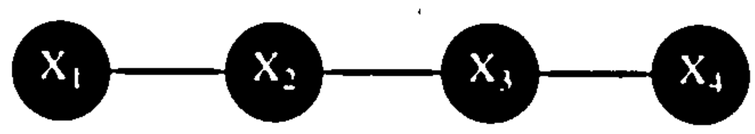



Figure 2.26: A Markov network which forms a chain.

Then by the recursive factorisation identity (chain rule):

$$P(x_1, x_2, x_3, x_4) = P(x_1)P(x_2 \mid x_1)P(x_3 \mid x_1, x_2)P(x_4 \mid x_1, x_2, x_3)$$

and using the conditional independence statements of the chain:

$$\begin{aligned} P(x_1, x_2, x_3, x_4) &= P(x_1)P(x_2 \mid x_1)P(x_3 \mid x_2)P(x_4 \mid x_3) \\ &= \frac{P(x_1)P(x_1, x_2)P(x_2, x_3)P(x_3, x_4)}{P(x_1)P(x_2)P(x_3)} \\ &= \frac{P(x_1, x_2)P(x_2, x_3)P(x_3, x_4)}{P(x_2)P(x_3)} \end{aligned}$$

This is equivalent to expressing the joint distribution as the product of the marginal distributions on the edges divided by the product of the intermediate nodes. But the edges of the chain graph are the cliques of the graph and the intermediate nodes, which we will call *separators*, are the intersections of neighbouring cliques. Thus if we represent $\mathcal{G}$ in the form of a chain of cliques with separators on the edges joining them as in Figure 2.27 then this chain will give us an easily accessible factorisation of $P(x_1, \ldots, x_4)$.

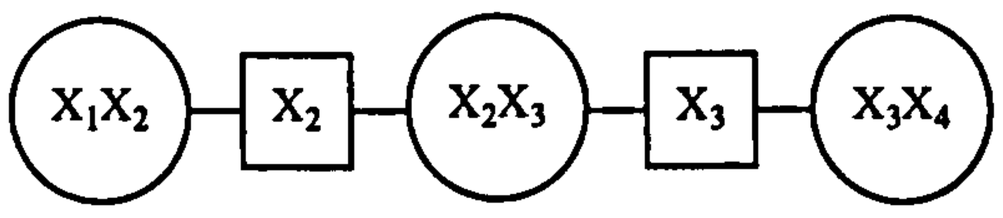



Figure 2.27: A clique chain.

Thus the clique chain is advantageous in determining the joint distribution of $X$ as a product of joint distributions of smaller state-space. It would be desirable if we could determine a structure derived from a weakly decomposable (and hence decomposable) graph for which this property will hold.

Pearl (1988) defines a *junction tree* or *join tree* for an undirected graph $\mathcal{G} = (K, E)$ as follows:

**Definition 16** *A junction tree $\mathcal{T}$ is one with the cliques of $\mathcal{G}$ as vertices, such that for every vertex $u$ of $\mathcal{G}$ , if we remove from $\dot{\mathcal{T}}$ all cliques not containing $u$, the remaining subtree remains connected. In other words, any two cliques containing $u$ are either adjacent in $\mathcal{T}$ or connected by a path made entirely of cliques that contain $u$.*

Related to this we define the *junction tree property*:

**Definition 17 Junction Tree Property:** *For any pair of cliques $C_1$ and $C_2$ of $\mathcal{G}$ then whenever $u \in C_1 \cap C_2$, then $u$ is contained in every clique on the path joining $C_1$ and $C_2$.*

We now present two theorems from Pearl (1988) which demonstrate the link between junction trees and weakly decomposable graphs.

**Theorem 8** *If an undirected graph $\mathcal{G}$ possesses a junction tree $\mathcal{T}$ then $\mathcal{G}$ is weakly decomposable.*

**Proof** Holds trivially if $\mathcal{G}$ consists of just one clique. Otherwise, suppose true for all proper subgraphs of $\mathcal{G}$. Take two cliques $C_1$ and $C_2$ adjacent in $\mathcal{T}$ and separate the link between them to form two subtrees $\mathcal{T}_1$ and $\mathcal{T}_2$. Let $\mathcal{G}_i$ be the union of the cliques in $\mathcal{T}_i$ $(i = 1, 2)$, the cliques in $\mathcal{T}_i$ are then the cliques in $\mathcal{G}_i$. $\mathcal{T}_i$ are then the junction trees for $\mathcal{G}_i$, which are proper subgraphs of $\mathcal{G}$, hence $\mathcal{G}_i$ are weakly decomposable. Suppose $u \in \mathcal{G}_1 \cap \mathcal{G}_2$, then there exists a clique $C_i'$ of $\mathcal{G}_i$ $(i = 1, 2)$ with $u \in C_i'$. Clearly the path in $\mathcal{T}$ joining $C_1'$ and $C_2'$ passes through both $C_1$ and $C_2$ therefore $u \in C_1 \cap C_2$, so $\mathcal{G}_1 \cap \mathcal{G}_2 \subseteq C_1 \cap C_2$. Since $C_1 \cap C_2 \subseteq \mathcal{G}_1 \cap \mathcal{G}_2$, $\mathcal{G}_1 \cap \mathcal{G}_2 = C_1 \cap C_2 = S$, say, is complete. Now take $u \in \mathcal{G}_1 \backslash S$, $v \in \mathcal{G}_2 \backslash S$, and suppose there exists a path $u, w_1, w_2, \ldots, w_k, v$ with each $w_i \notin S$. There exists a clique $C$ containing the complete set $\{u, w_1\}$. Clearly $C \subseteq \mathcal{G}_1$, so $w_1 \in \mathcal{G}_1$, hence $w_1 \in \mathcal{G}_1 \backslash S$. Repeat the argument to deduce $w_2 \in \mathcal{G}_1 \backslash S, \ldots, v \in \mathcal{G}_1 \backslash S$, which is a contradiction. So $S$ separates $\mathcal{G}_1 \backslash S$ from $\mathcal{G}_2 \backslash S$, and $(\mathcal{G}_1 \backslash S, \mathcal{G}_2 \backslash S, S)$ is a weak decomposition of $\mathcal{G}$.

□

**Theorem 9** *If $\mathcal{G}$ is weakly decomposable, then $\mathcal{G}$ possesses a junction tree $\mathcal{T}$.*

**Proof** Trivial if $\mathcal{G}$ is a single clique. Otherwise suppose true for proper subgraphs of $\mathcal{G}$. Let $(\mathcal{G}_1 \backslash S, \mathcal{G}_2 \backslash S, S)$, where $S = \mathcal{G}_1 \cap \mathcal{G}_2$, be a weak decomposition of $\mathcal{G}$, then at least one of $\mathcal{G}_1$ and $\mathcal{G}_2$, say $\mathcal{G}_1$, has the form $\cup\{C : C \in \mathcal{C}_1\}$, with $\mathcal{C}_1 \subset \mathcal{C}$ where $\mathcal{C}$ is the set of cliques in $\mathcal{T}$; and then we may redefine $\mathcal{G}_2 = \cup\{C : C \in \mathcal{C}_2\}$ $(\mathcal{C}_2 = \mathcal{C} \backslash \mathcal{C}_1)$ and still have $(\mathcal{G}_1 \backslash S, \mathcal{G}_2 \backslash S, S)$ a weak decomposition. Let $C_i \in \mathcal{C}_i$ satisfy $S \subseteq C_i$. By hypothesis we have a junction tree $\mathcal{T}_i$ for $\mathcal{G}_i$. $\mathcal{T}$ may be formed by linking $C_1 \in \mathcal{T}_1$ to $C_2 \in \mathcal{T}_2$. Let $u \in K$. If $u \notin \mathcal{G}_2$, then all cliques containing $u$ are in $\mathcal{G}_1$, and so are connected in $\mathcal{T}_1$, and hence in $\mathcal{T}$. If $u \notin \mathcal{G}_1$, similarly. Otherwise if $u \in \mathcal{G}_1$ and $u \in \mathcal{G}_2$ then $u \in S$. The cliques in $\mathcal{C}_i$ containing $u$ are connected in $\mathcal{T}_i$ and include $C_i$. Since $C_1$ and $C_2$ are connected in $\mathcal{T}$ the result follows.

□

The following algorithm (adapted from Pearl, 1988 and Golumbic, 1980) may be used to assemble the junction tree of an undirected graph:

**Definition 18 An Algorithm to Assemble a Junction Tree**

*i) Assume that $\mathcal{G} = (K, E)$ is an undirected graph with $k$ vertices and that $\mathcal{G}$ is connected or we can apply the algorithm to connected subgraphs of $\mathcal{G}$. Generate a chordal graph $\mathcal{G}'$ from $\mathcal{G}$ ( if $\mathcal{G}$ is chordal then $\mathcal{G}' = \mathcal{G}$).*

*ii) Choose an arbitrary vertex of $\mathcal{G}'$ and label it 1. For $i = 2, 3, \ldots, k$ repeat the following algorithm: Let $L_i$ be the set of vertices already labelled. Let $U_i = K \backslash L_i$ be the vertices which have not yet been labelled. Give label $i$ to the vertex in $U_i$ which has the most neighbours already labelled, with an arbitrary choice between vertices that tie on this criterion. Let $D_i$ be the neighbours of $i$ with lower labels. Since $\mathcal{G}'$ is triangulated $L_i \cap D_i$ will be a complete subset.*

*iii) Let $\pi_i$ be the number of vertices in $L_i \cap D_i$. Vertex $i$ is termed a* **ladder node** *if $i = k$ or if $i < k$ and $\pi_{i+1} < 1 + \pi_i$. There is a $1:1$ correspondence between the ladder nodes and the cliques. The $j$th clique consists of the $j$th ladder node, $i$ say, together with $L_i \cap D_i$. Therefore the maximum number of cliques is $k$. Assume we have $t$ cliques.*

*iv) Order the cliques $C_1, C_2, \ldots, C_t$ by rank of the highest labelled vertex in each clique.*

*v) Form the junction tree by connecting each $C_j$ to a predecessor $C_l (l > j)$ sharing the highest number of vertices with $C_j$.*

Consider the chordal graph of Figure 2.28 below. We determine its cliques to be $(A, B, C)$ , $(B, C, D)$ , and $(C, E)$. We may thus construct the junction trees of Figure 2.29 from it by the above algorithm. Hence junction trees are shown not to be unique.

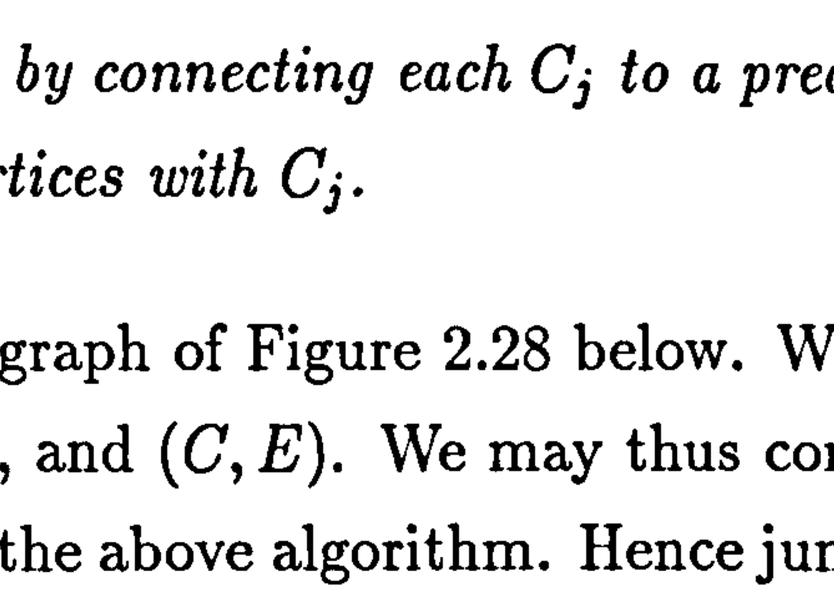



Figure 2.28: A chordal graph.

Now let us consider the separation properties of the junction tree.

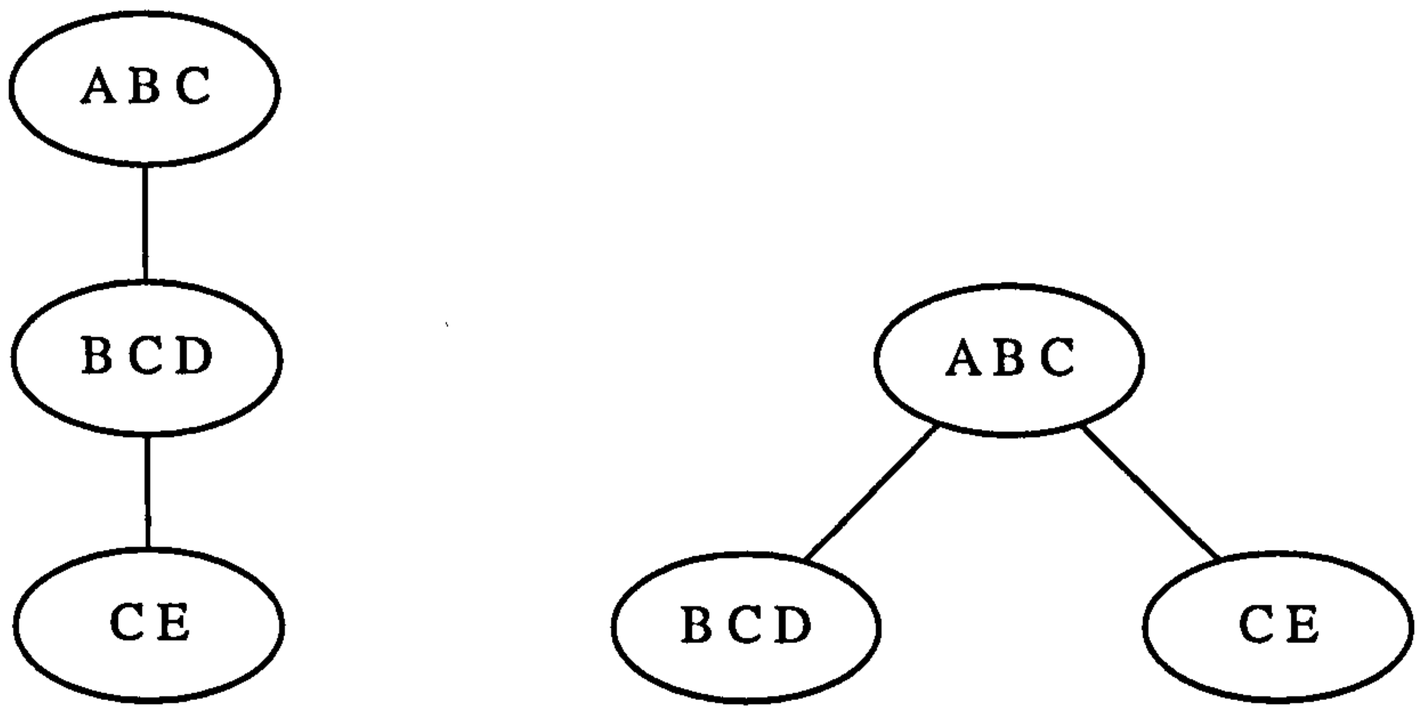



Figure 2.29: Two possible junction trees for the chordal graph of Figure 2.28.

**Theorem 10** *If a separator $S$ is removed from a connected junction tree $T$, producing two connected subtrees $\mathcal{T}_A$ and $\mathcal{T}_B$, containing variables $A$ and $B$, say, then $A \perp\!\!\!\perp B \mid S$.*

**Proof** Consider any two nodes $u$ and $v$ in cliques $C_1$ and $C_2$, say, of $\mathcal{T}_A$ and $\mathcal{T}_B$ respectively. Then by construction of $\mathcal{T}_A$ and $\mathcal{T}_B$, $S$ joins two cliques $C_A$ and $C_B$, say, in $\mathcal{T}_A$ and $\mathcal{T}_B$ respectively and $S = A \cap B = C_A \cap C_B$. Now, since $\mathcal{T}_A$ and $\mathcal{T}_B$ are both connected subtrees of $\mathcal{T}$ there must exist paths between $C_1$ and $C_A$ and between $C_B$ and $C_2$. Hence there is a path between $u$ and $v$ which passes through $S$. See Figure 2.30.

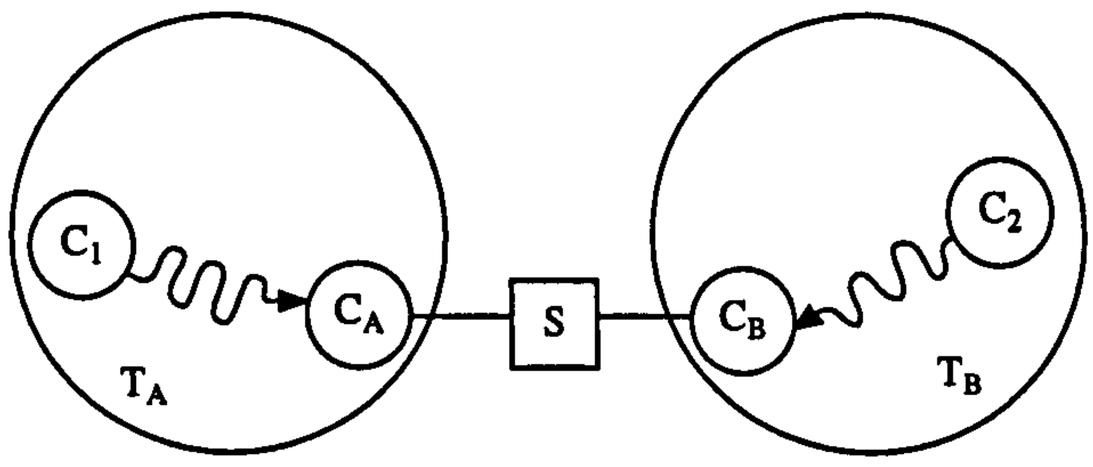



Figure 2.30: A junction tree to illustrate Theorem 10.

Now let us assume that a path exists from $u$ to $v$ which does not pass through $S$. Then on our path between $u$ and $v$ we must have cliques $C'_A \subseteq \mathcal{T}_A$ and $C'_B \subseteq \mathcal{T}_B$ , say, joined by a separator $S' = C'_A \cap C'_B$. See Figure 2.31.

We thus have two paths from $u$ to $v$: one which passes through $S'$ but not through $S$, and one which passes through $S$ but not through $S'$. But this means we have a loop in our tree structure with both $S$ and $S'$ joining the connected subtrees $\mathcal{T}_A$ and $\mathcal{T}_B$ which contradicts our tree structure. Thus every path that exists between any two nodes $u \in A$ and $v \in B$ must pass through $S = A \cap B$, so $A \perp\!\!\!\perp B \mid S$.

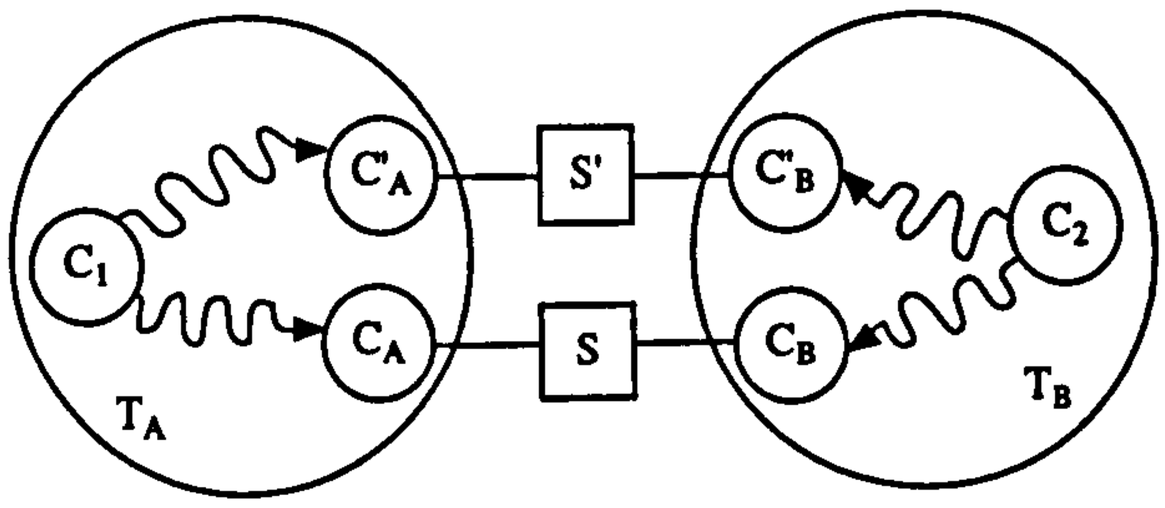



Figure 2.31: A second junction tree to illustrate Theorem 10.

□

**Theorem 11 Separation Theorem:** *Consider a junction tree $\mathcal{T}$ with neighbouring cliques $C_i$ and $C_j$ joined by a separator $S = C_i \cap C_j$, then:*

$$C_i \perp\!\!\!\perp C_j \mid S$$

**Proof** Recall from Theorem 10 that if a separator $S$ is removed from a connected junction tree $\mathcal{T}$ producing two connected subtrees $\mathcal{T}_A$ and $\mathcal{T}_B$, containing variables $A$ and $B$, say, then $A \perp\!\!\!\perp B \mid S$. Thus if $S = C_i \cap C_j$, then either $C_i \subseteq A$ and $C_j \subseteq B$ or $C_i \subseteq B$ and $C_j \subseteq A$. Without loss of generality let $C_i \subseteq A$ and $C_j \subseteq B$ then using the reduction lemma:

$$\begin{aligned} & A \perp\!\!\!\perp B \mid S \\ \Rightarrow \quad & (C_i, A \backslash C_i) \perp\!\!\!\perp (C_j, B \backslash C_j) \mid S \\ \Rightarrow \quad & C_i \perp\!\!\!\perp C_j \mid S \end{aligned}$$

□

**Theorem 12** *If $\mathcal{T}$ is a junction tree with cliques $C \subseteq \mathcal{C}$ and separators $S \subseteq \mathcal{S}$, formed from a weakly decomposable graph $\mathcal{G} = (K, E)$ for a set of random variables $X = (X_1, X_2, \ldots, X_k)$, then the joint distribution function of $X$ is the product of the marginal distribution functions of the cliques divided by the marginal distribution functions of the separators.*

**Proof** We showed in Theorem 10 that if a separator $S$ is removed from a connected junction tree $\mathcal{T}$ producing two connected subtrees $\mathcal{T}_A$ and $\mathcal{T}_B$, containing variables $A$ and $B$, say, then $A \perp\!\!\!\perp B \mid S$. We may therefore express the joint distribution function $f_X(x)$ of $X$ as follows:

$$f_X(x) = \frac{f_A(x_A) f_B(x_B)}{f_S(x_S)}$$

We may repeat this logic on $\mathcal{T}_A$ and $\mathcal{T}_B$ and any subtrees formed thereafter until all the separators have been removed and the $t$ subtrees formed consist of a single clique each. Each time a separator is removed from a subtree the denominator of our factorisation of $f_X(x)$ is multiplied by the marginal distribution function of the separator, and a term in the numerator becomes a product of two marginal distributions. Thus the numerator in our factorisation of $f_X(x)$ will eventually consist of the product of the marginals on all the cliques, and the denominator will consist of the product of the marginals on all the separators, i.e.:

$$f_X(x) = \frac{\prod_{C_i \subseteq \mathcal{C}} f_{C_i}(x_{C_i})}{\prod_{S_j \subseteq \mathcal{S}} f_{S_j}(x_{S_j})}$$

□

So junction trees may be formed from weakly decomposable (and hence decomposable) graphs and they provide a simple factorisation of the joint probability density function of the random variables they represent. We now introduce a property particular to junction trees formed from a strongly decomposable graph.

**Theorem 13** *A junction tree $\mathcal{T}$ formed from a strongly decomposable graph $\mathcal{G} = (K, E)$ has at least one* **strong root.** *Where a clique $R$ on a junction tree is termed a strong root if any pair $A$, $B$ of neighbouring cliques on the tree with $A$ closer to $R$ than $B$ satisfies:*

$$(B \backslash A) \subseteq \Gamma \vee (B \cap A) \subseteq \Delta$$

For proof see Leimer (1989).

## 2.12 Propagation in a Junction Tree

We have shown how we may form a junction tree, $\mathcal{T}$, from an original graph $\mathcal{G}$ and that for such a tree the joint distribution on all the variables may be expressed as the product of the joint distributions on the cliques divided by the product of the joint distributions on the separators. We now consider a strategy which will ensure that our junction tree holds this representation. Lauritzen & Spiegelhalter (1988) present such a scheme defined explicitly for discrete variables. An object-orientated version of their scheme may be found in Jensen *et al.* (1990), and a generalisation of this method is presented in Dawid (1992). The approach we present here makes the simple generalisation of allowing mixed networks and hence includes the two pure cases.

### 2.12.1 Propagation

Suppose we have an independence graph $\mathcal{G} = (K, E)$ for a set of random variables $X = (X_1, X_2, \ldots, X_k)$ such that each vertex $a \in K$ corresponds to a random variable $X_a$. Let us partition $K$ into a set of discrete vertices $\Delta$ and a set of continuous vertices $\Gamma$ such that $K = \Delta \cup \Gamma$. Let $I_\delta$, for $\delta \in \Delta$, denote a particular discrete variable $X_\delta$ and let $Y_\gamma$, for $\gamma \in \Gamma$, denote a particular continuous variable $X_\gamma$. Then $I_\delta$ takes values in a discrete space $\mathcal{X}_\delta = \mathcal{I}_\delta$ and $Y_\gamma$ takes real values in a continuous space $\mathcal{X}_\gamma = I\!R$. For any $A \subseteq K$ we write $\mathcal{X}_A$ for the space $\times_{a \in A} \mathcal{X}_a$ and in particular:

$$\mathcal{X}_A = \times_{a \in A} \mathcal{X}_a = \mathcal{I}_A \times \mathcal{Y}_A = (\times_{\delta \in A \cap \Delta} \mathcal{X}_\delta) \times (\times_{\gamma \in A \cap \Gamma} \mathcal{X}_\gamma)$$

We abbreviate $\mathcal{X}_K$ to $\mathcal{X}$. If $x = (x_a : a \in K)$, then we let $x_A = (x_a : a \in A)$. A typical component $x$ of the joint state space of discrete and continuous variables $\mathcal{X}$ may be written in terms of its discrete and continuous components thus:

$$x = (x_a)_{a \in K} = (i, y) = \left((i_\delta)_{\delta \in \Delta}, (y_\gamma)_{\gamma \in \Gamma}\right)$$

We will similarly write a typical component $x_A$ of the state space $\mathcal{X}_A$ as:

$$x_A = (x_a)_{a \in A} = (i_A, y_A) = \left((i_\delta)_{\delta \in A \cap \Delta}, (y_\gamma)_{\gamma \in A \cap \Gamma}\right)$$

Associated with $\mathcal{G}$ we may determine a junction tree $\mathcal{T}$ possibly by moralising and weakly triangulating $\mathcal{G}$ first. Let $\mathcal{T}$, have vertex-set $\mathcal{C}$ and edge-set $\mathcal{S}$. Associated with any $C \in \mathcal{C}$ is a subset of $K$, which we will also denote $C$. Such subsets are the *cliques* of the junction tree as defined earlier. The union of all the cliques is $K$. Associated with an edge $S \in \mathcal{S}$ is a subset of $K$, which we will also denote by $S$ and term a *separator*. The separator which joins any two cliques $C$ and $C'$ is $S = C \cap C'$. We collectively term cliques and separators *universes*. The joint density (probability mass function) $f_K$ of the random variables $X$ may, by virtue of the factorisation criterion, be written in the form:

$$f_K(x) = \prod_{C \in \mathcal{C}} a_C(x_C) \tag{2.2}$$

where each $a_C$ is a known non-negative real function on $\mathcal{X}_C$. If we have non-negative real functions $b_S$ on $\mathcal{X}_S$ such that $b_S \equiv 1$ then we may trivially write:

$$f_K(x) = \frac{\prod_{C \in \mathcal{C}} a_C(x_C)}{\prod_{S \in \mathcal{S}} b_S(x_S)} \tag{2.3}$$

We will term any collection of non-negative real functions $\mathcal{K} = (\{a_C : C \in \mathcal{C}\}, \{b_S : S \in \mathcal{S}\})$ on the various cliques and separators a *charge* on $\mathcal{T}$. The individual functions $a_C$ and $b_S$ are termed *potentials*. When Equation 2.3 holds we call $\mathcal{K}$ a *representation* for $f$ on $\mathcal{T}$. For any charge, the expression on the right-hand side of Equation 2.3 is interpreted as 0 if the denominator is 0 and the function is then termed the *contraction* of $\mathcal{K}$. A non-negative function $f$ is said to *factorise* on $\mathcal{T}$ if and only if there exists a representation for $f$ on $\mathcal{T}$.

We shall also write Equation 2.3 more generally as:

$$f = \frac{\prod_{C \in \mathcal{C}} a_C}{\prod_{S \in \mathcal{S}} b_S} \tag{2.4}$$

where we do not restrict $f$ to be a density.

The first phase of our probability propagation algorithm is the *initialisation* phase. We may *initialise* the system by arbitrarily defining the potentials $a_C$ such that they match the form of Equation 2.2 and then, if we put $b_S \equiv 1$, $\mathcal{K}$ will be a representation. The second phase is the *propagation* phase. In this phase we wish to determine a way in which we may alter our potentials so that they each become equal to the marginal density of the variables they represent. We will also require the overall joint density to remain the same. Four basic operations (*extension*, *multiplication*, *division*, and *marginalisation*) will be required to form our propagation algorithm. We will define these as follows:

**Definition 19 Extension:** *Let $U \subseteq V \subseteq K$ and $\phi(x_U) = \phi(i_U, y_U)$ be a potential function defined on $\mathcal{X}_U = \mathcal{I}_U \times \mathcal{Y}_U$. Let $\eta(x_V)$ be the* **extension** *of $\phi(x_U)$ to $V$, where $\eta(x_V)$ is a potential function defined on $\mathcal{X}_V = \mathcal{I}_V \times \mathcal{Y}_V = (\mathcal{I}_U \times \mathcal{I}_{V \setminus U}) \times (\mathcal{Y}_U \times \mathcal{Y}_{V \setminus U})$, if:*

$$\eta(x_V) = \eta(i_U, i_{V \setminus U}, y_U, y_{V \setminus U}) = \phi(i_U, y_U) = \phi(x_U)$$

Where it is clear from the context we need not distinguish between a potential function and its extension.

**Definition 20 Multiplication:** *Let $\phi$ and $\eta$ be two potential functions defined on the spaces $\mathcal{X}_U$ and $\mathcal{X}_V$ respectively with $U \subseteq K$ and $V \subseteq K$. Then the* **multiplication** *of $\phi$ and $\eta$, denoted $\phi \times \eta$, defined on the space $\mathcal{X}_{U \cup V}$ is:*

$$(\phi \times \eta)(x_{U \cup V}) = \phi(x_{U \cup V})\, \eta(x_{U \cup V})$$

*where $\phi$ and $\eta$ on the right-hand side have first been extended to occupy $\mathcal{X}_{U \cup V}$.*

**Definition 21 Division:** *Let $\phi$ and $\eta$ be two potential functions defined on the spaces $\mathcal{X}_U$ and $\mathcal{X}_V$ respectively with $U \subseteq K$ and $V \subseteq K$. Then the* **division** *of $\phi$ by $\eta$, denoted $\phi/\eta$, defined on the space $\mathcal{X}_{U \cup V}$ is:*

$$(\phi/\eta)(x_{U\cup V}) = \begin{cases} (\phi(x_{U\cup V})/\eta(x_{U\cup V})) & \text{if } \eta(x_{U\cup V}) \neq 0 \\ 0 & \text{if } \eta(x_{U\cup V}) = 0 \end{cases}$$

*where $\phi$ and $\eta$ on the right-hand side have first been extended to occupy $\mathcal{X}_{U\cup V}$.*

In general, where we may avoid confusion, we will use the summation operator $\sum$ as a shorthand notation for the process of *marginalisation* which may in fact involve both integration and summation. We thus define marginalisation as follows:

**Definition 22 Marginalisation:** *Let $U \subseteq V \subseteq K$ and $\phi(x_V) = \phi(i_V, y_V) = \phi(i_U, i_{V\setminus U}, y_U, y_{V\setminus U})$ be a potential function defined on $\mathcal{X}_V = \mathcal{I}_V \times \mathcal{Y}_V = (\mathcal{I}_U \times \mathcal{I}_{V\setminus U}) \times (\mathcal{Y}_U \times \mathcal{Y}_{V\setminus U})$. Then we will let the expression $\sum_{V\setminus U} \phi(x_V)$ denote the* **marginalisation** *of $\phi(x_V)$, with respect to $(I_{V\setminus U}, Y_{V\setminus U})$, to a function $\eta(x_U)$ defined on the space $\mathcal{X}_U$ where:*

$$\eta(x_U) = \sum_{V\setminus U} \phi(x_V) = \int_{y_{V\setminus U}} \sum_{i_{V\setminus U}} \phi(x_V)\, dy_{V\setminus U}$$

We will define a function $f_A$ as follows:

$$f_A = \sum_{K\setminus A} f$$

As stated earlier we need not restrict $f$ to be a density. In particular, we shall not require $\sum_K f = 1$. When $f$ is a joint density for $X$, however, then $f_A$ is the implied joint density of $X_A$. $f_A$ will, in general, be termed the *sum-margin*, or *margin*, of $f$ on $A$.

The propagation algorithm Jensen *et al.* present consists of a series of *flows* between neighbouring cliques in the junction tree. Each flow affects the potentials on exactly one clique and one separator and is composed of the basic operations just discussed.

**Definition 23** *Let $C_1$ and $C_2$ be adjacent cliques in a junction tree $\mathcal{T}$ which are joined by a separator $S_0$. Then a* **flow** *passing from $C_1$ (the* **source***) to $C_2$ (the* **sink***) replaces an original charge $\mathcal{K} = (\{a_C : C \in \mathcal{C}\}, \{b_S : S \in \mathcal{S}\})$, by a new charge $\mathcal{K}^* = (\{a^*_C : C \in \mathcal{C}\}, \{b^*_S : S \in \mathcal{S}\})$ , where:*

$$
\begin{aligned}
b^*_{S_0} &= \sum_{C_1 \setminus S_0} a_{C_1} \\
a^*_{C_2} &= a_{C_2} \times \lambda_{S_0} \\
\lambda_{S_0} &= \begin{cases} b^*_{S_0}(x_{S_0}) / b_{S_0}(x_{S_0}) & if\, b_{S_0}(x_{S_0}) > 0 \\ 0 & if\, b_{S_0}(x_{S_0}) = 0 \end{cases}
\end{aligned}
\tag{2.5}
$$

*and all other potentials are unaltered.*

In line with Dawid (1992) this flow is also known as a *sum-flow* since the marginalisation operator is used. $\lambda_{S_0}$ is termed the *update factor* carried by the flow along $S_0$ to $C_2$.

**Lemma 4** *The passage of a flow does not affect the contraction of a charge.*

**Proof** Consider passing a flow between two adjacent cliques $C_1$ and $C_2$ via a separator $S_0$. Let $f$ and $f^*$ be the respective contractions of $\mathcal{K}$ and $\mathcal{K}^*$, the charges before and after the passage of the flow. Then, since the passage of the flow will, by definition, only affect the potential functions $a_{C_2}$ and $b_{S_0}$, in order to show that $f^*(x) = f(x)$ for all $x \in \mathcal{X}$ we must show that $a_{C_2}/b_{S_0} = a^*_{C_2}/b^*_{S_0}$. We may distinguish three cases:

(i) $b_{S_0} > 0$ and $b^*_{S_0} > 0$. In this case:

$$
\frac{a^*_{C_2}}{b^*_{S_0}} = \frac{a_{C_2} \times \lambda_{S_0}}{b^*_{S_0}} = \frac{a_{C_2} \times \left(\frac{b^*_{S_0}}{b_{S_0}}\right)}{b^*_{S_0}} = \frac{a_{C_2}}{b_{S_0}}
$$

so $f^*(x) = f(x)$ for all $x \in \mathcal{X}$.

(ii) $b_{S_0} > 0$ and $b^*_{S_0} = 0$. Since $a_{C_1}$ is non-negative if $b^*_{S_0} = \sum_{C_1 \setminus S_0} a_{C_1} = 0$ then $a_{C_1} = 0$, hence $f(x) = 0$ for all $x \in \mathcal{X}$. But, since $b^*_{S_0} = 0$ following the definitions of the respective operators:

$$
\frac{a^*_{C_2}}{b^*_{S_0}} = \frac{a_{C_2} \times \left(\frac{b^*_{S_0}}{b_{S_0}}\right)}{b^*_{S_0}} = \frac{a_{C_2} \times \left(\frac{0}{b_{S_0}}\right)}{0} = \frac{a_{C_2} \times 0}{0} = \frac{0}{0} = 0
$$

so $f^*(x) = 0$ also. Thus $f^*(x) = f(x)$ for all $x \in \mathcal{X}$.

(iii) $b_{S_0} = 0$. Thus $a_{C_2}/b_{S_0} = 0$, by definition of division, and so $f(x) = 0$ for all $x \in \mathcal{X}$. Similarly:

$$\frac{a^*_{C_2}}{b^*_{S_0}} = \frac{a_{C_2} \times \left(\frac{b^*_{S_0}}{b_{S_0}}\right)}{b^*_{S_0}} = \frac{a_{C_2} \times 0}{b^*_{S_0}} = \frac{0}{b^*_{S_0}} = 0$$

so $f^*(x) = 0$ also. Thus $f^*(x) = f(x)$ for all $x \in \mathcal{X}$.

□

We now define an operation termed *absorption* which consists of a series of flows.

**Definition 24** *Suppose we have a junction tree $\mathcal{T}$ with cliques $\mathcal{C}$ and separators $S$. Let $C \in \mathcal{C}$ and let $C_1, \ldots C_m$ be a collection of $m$ neighbours of $C$ joined to $C$ by separators $S_1, \ldots, S_m$ respectively. Then $C$ is said to* **absorb** *from $C_1, \ldots, C_m$ if a flow is passed from each $C_i$ to $C$ through $S_i$ for all $i = 1, \ldots m$.*

A *schedule* is an ordered list of directed edges on $\mathcal{T}$, specifying which flows are to be passed and in what order. A flow is termed *active* if before it is sent the source has itself received active flows from all its neighbours in $\mathcal{T}$, with the possible exception of the sink, and it is the first flow in the list with this property. Using this definition the first active flow must originate in a leaf of $\mathcal{T}$. A schedule is termed *active* if it contains only active flows. A schedule is *full* if it contains an active flow in each direction along every edge of $\mathcal{T}$. The schedule is *fully active* if it is both full and active. A fully active schedule may be obtained from any full schedule by omitting all *inactive* flows.

**Proposition 6** *For any tree $\mathcal{T}$, there exists a fully active schedule.*

**Proof** If $\mathcal{T}$ consists of a single clique then the result is vacuously true. Otherwise we may form a fully active schedule for $\mathcal{T}$ if we apply the following strategy:

(i) Start with an empty schedule and a connected tree $\mathcal{T}$ with $n$ separators, say. Let $l = 1$ and $\mathcal{T}_l = \mathcal{T}$.

(ii) Select a leaf from $\mathcal{T}_l$. Let $C_{0,l}$ denote the leaf, $S_{0,l}$ denote the edge joining $C_{0,l}$ to the rest of $\mathcal{T}_l$ and $C_{1,l}$ denote the clique it is joined to.

(iii) Let the $l-$th flow in the schedule be from $C_{0,l}$ to $C_{1,l}$ via $S_{0,l}$, and the $(n + 1 - l)-$th flow in the schedule be from $C_{1,l}$ to $C_{0,l}$ via $S_{0,l}$.

(iv) Drop $C_{0,l}$ and $S_{0,l}$ from $\mathcal{T}_l$ to form a subtree $\mathcal{T}_{l,0}$ of $\mathcal{T}_l$.

(v) If $\mathcal{T}_{l,0}$ consists of a single clique then end. Otherwise let $l = l+1$, $\mathcal{T}_l = \mathcal{T}_{l-1,0}$ and repeat the algorithm from step (ii).

Since there are $n$ separators in $\mathcal{T}$ and each stage of our algorithm removes one separator and adds two flows there are $n$ stages in the algorithm and $2n$ flows in the schedule thus formed. Consider a stage $l$ in which we have a tree $\mathcal{T}_l$ with a leaf $C_{0,l}$ joined to a clique $C_{1,l}$ by an edge $S_{0,l}$. Assume that there exists a fully active schedule for the subtree $\mathcal{T}_{l,0}$ formed as in step (iv). Then, by assumption, we may form a fully active subtree for $\mathcal{T}_l$ by adding a flow from $C_{0,l}$ to $C_{1,l}$ via $S_{0,l}$ at the beginning of this schedule, and a flow from $C_{1,l}$ to $C_{0,l}$ via $S_{0,l}$ at the end of this schedule. At the end of the final stage $n$, the tree $\mathcal{T}_{n,0}$ is a single clique. The schedule for this clique is empty and hence fully active. Thus by induction we may form a fully active schedule for $\mathcal{T}$. We notice that the flows we need to add form the same order as described in our algorithm.

□

The algorithm in the proof of Proposition 6 defines a fully active schedule which may be termed *palindromic* in that it comprises two schedules each consisting of $n$ flows with the second schedule being an exact reversal of the first in both order and direction. Jensen *et al.* (1990) propose a schedule which defines two phases- a *collect evidence* phase and a *distribute evidence* phase. In the collect evidence phase active flows are passed along edges which are directed towards some arbitrary root-clique $C^* \in \mathcal{C}$. In the distribute evidence phase starting from $C^*$ active flows are passed back towards the periphery. Notice that the two schedules of our palindromic schedule may be used to define the collect evidence and distribute evidence phases of propagation.

Let $\mathcal{T}'$ be a subtree of $\mathcal{T}$, with vertices $\mathcal{C}' \subseteq \mathcal{C}$ and edges $\mathcal{S}' \subseteq \mathcal{S}$. The *base* of $\mathcal{T}'$ is defined to be $K' = \cup\{C : C \in \mathcal{C}'\}$, the collection of variables associated with $\mathcal{T}'$. If $\mathcal{K} = (\{a_C : C \in \mathcal{C}\}, \{b_S : S \in \mathcal{S}\})$ is a charge for $\mathcal{T}$ then its *restriction* to $\mathcal{T}'$ is $\mathcal{K}_{\mathcal{T}'} = (\{a_C : C \in \mathcal{C}'\}, \{b_S : S \in \mathcal{S}'\})$, and its potential on $\mathcal{T}'$ is the contraction of $\mathcal{K}_{\mathcal{T}'}$. With respect to a given schedule of flows, a subtree $\mathcal{T}'$ is said to be *live* at a certain stage of the schedule if it has already received active flows from all its neighbours. This property is then, obviously, retained throughout the remainder of the schedule.

**Theorem 14** *Suppose that we start with a representation* $\mathcal{K}^0 = (\{a^0_C : C \in \mathcal{C}\}, \{b^0_S : S \in \mathcal{S}\})$ *for a function* $f$ *which factorises on* $\mathcal{T}$*, and progressively modify*

*it by passing a sequence of flows according to some schedule. Then whenever $\mathcal{T}'$ is live, the potential on $\mathcal{T}'$ is the margin $f_{K'}$ of $f$ on $K'$.*

**Proof** By Lemma 4 we have a representation for $f$ on $\mathcal{T}$ at all stages during the passage of the schedule. If $\mathcal{T}' = \mathcal{T}$ then, since $f$ factorises on $\mathcal{T}$, the potential on $\mathcal{T}'$ is $f = f_K = f_{K'}$. Otherwise, suppose that $\mathcal{T}'$ is a subtree of $\mathcal{T}$ such that $\mathcal{T}' \neq \mathcal{T}$.

Let $\mathcal{T}'$ be live and let $C^*$ be the last neighbour of $\mathcal{T}'$ to have passed a flow into $\mathcal{T}'$. We shall assume that $C^*$ is joined to $\mathcal{T}'$ by a separator $S^*$. Let $\mathcal{T}^*$ be the subtree of $\mathcal{T}$ obtained by adding $C^*$ and $S^*$ to $\mathcal{T}'$. We shall let $\mathcal{C}^*$, $\mathcal{S}^*$ and $K^*$ denote the cliques, separators and base of $\mathcal{T}^*$. By the junction tree property $S^* = C^* \cap K'$ so $\mathcal{C}^* = \mathcal{C}' \cup \{C^*\}$, $\mathcal{S}^* = \mathcal{S}' \cup \{S^*\}$, and $K^* = K' \cup C^*$. Let $\mathcal{K}' = (\{a'_C : C \in \mathcal{C}\}, \{b'_S : S \in \mathcal{S}\})$ be the representation for $f$ just before the last flow from $C^*$ into $\mathcal{T}'$.

Assume the result is true for $\mathcal{T}^*$. Then just before the last flow from $C^*$ into $\mathcal{T}'$:

$$f_{K^*} = \frac{a_{C^*} \times \prod_{C \in \mathcal{C}'} a_C}{b_{S^*} \times \prod_{S \in \mathcal{S}'} b_S}$$

is the the margin of $f$ on $K^*$. So, by Definition 23, just after the last flow from $C^*$ into $\mathcal{T}'$ the potential on $\mathcal{T}'$ is:

$$\frac{\lambda_{S^*} \times \prod_{C \in \mathcal{C}'} a_C}{\prod_{S \in \mathcal{S}'} b_S}$$

where $\lambda_{S^*}$ is the update factor. But since the margin of $f$ on $K'$ is:

$$\begin{aligned} \sum_{C^* \setminus S^*} f_{K^*} &= \frac{\prod_{C \in \mathcal{C}'} a_C \times \sum_{C^* \setminus S^*} a_{C^*}}{\prod_{S \in \mathcal{S}'} b_S \times b_{S^*}} \\ &= \frac{\lambda_{S^*} \times \prod_{C \in \mathcal{C}'} a_C}{\prod_{S \in \mathcal{S}'} b_S} \end{aligned}$$

the potential on $\mathcal{T}'$ is the margin of $f$ on $K'$, so the result is true by induction.

□

If we consider a directed edge $C \to C'$ where $C$ and $C'$ are adjacent cliques in $\mathcal{T}$ connected by a separator $S$, then removing $S$ from $\mathcal{T}$ leaves two disjoint subtrees, the *tail* and *head* of $S$ containing $C$ and $C'$ respectively.

**Corollary 1** *Let $S$ be a directed edge out of a clique $C$, having head $\mathcal{T}^+$ with base $K^+$, and let $K^* = C \cup K^+$. Then at any time after $C$ has received active flows from all its neighbours in the tail $\mathcal{T}^-$ of $S$, the potential on $C$ satisfies:*

$$\frac{a_C}{\left(\sum_{C \setminus S} a_C\right)} = \frac{f_C}{f_S} = \frac{f_{K^*}}{f_{K^+}} \tag{2.6}$$

**Proof** $C$ has received active flows from all its neighbours in $\mathcal{T}^-$ so $\mathcal{T}^*$ is live and, by Theorem 14, the potential on $\mathcal{T}^*$ is the margin of $f$ on $K^*$ which may be written as:

$$f_{K^*} = \frac{a_C \times \prod_{C' \in C^+} a_{C'}}{b_S \times \prod_{S' \in S^+} b_{S'}}$$

$\mathcal{T}^+$ will be live following the passage of a further flow out of $C$ and along $S$. The potential on $\mathcal{T}^+$ will then be the margin of $f$ on $K^+$ which may be written as:

$$f_{K^+} = \frac{\lambda_S \times \prod_{C' \in C^+} a_{C'}}{\prod_{S' \in S^+} b_{S'}}$$

Hence:

$$\frac{f_{K^*}}{f_{K^+}} = \frac{a_C/b_S}{\lambda_S} = \frac{a_C/b_S}{\sum_{C \setminus S} a_C/b_S} = \frac{a_C}{\sum_{C \setminus S} a_C} \tag{2.7}$$

But, by Theorem 10, since $S$ separates $C \setminus S$ from $\mathcal{T}^* \setminus S$ we may apply the factorisation criterion for conditional independence (Proposition 3) to $f_{K^*}$ and hence:

$$f_{K^*} = f_{S,C \setminus S,K^+ \setminus S} = \frac{f_{S,C \setminus S} f_{S,K^+ \setminus S}}{f_S} = \frac{f_C f_{K^+}}{f_S} \tag{2.8}$$

Thus putting Equations 2.7 and 2.8 together we obtain Equation 2.6.

□

**Corollary 2** *Whenever a clique is live, its potential is $f_C$.*

**Proof** Follows directly from Theorem 14 if we put $\mathcal{T}' = C$.

□

**Corollary 3** *Any time after active flows have passed in both directions across an edge in $\mathcal{T}$, the potential for the associated separator is $f_S$.*

**Proof** Consider two subtrees $\mathcal{T}'$ and $\mathcal{T}^*$ of $\mathcal{T}$ for which $S = \mathcal{T}' \cap \mathcal{T}^*$. Let $C'$ and $C^*$ be the cliques in $\mathcal{T}'$ and $\mathcal{T}^*$ respectively which are joined by $S$. Without loss of generality let us assume that the first active flow across $S$ passes from $C'$ to $C^*$ then $C^*$ will have received active flows from its neighbour $C'$ in $\mathcal{T}'$. Just before the second active flow passes across $S$, this time from $C^*$ to $C'$, $C^*$ must have received active flows from all its neighbours in $\mathcal{T}^*$. Therefore $C^*$ has received active flows from all its neighbours in $\mathcal{T}$. In other words $C^*$ is live. But by Corollary 2 since $C^*$ is live its potential is $f_{C^*}$. Thus when the second active flow is passed across $S$ the potential $b_S$ on $S$ is:

$$b_S = \sum_{C^* \setminus S} f_{C^*} = f_S$$

Hence the result.

□

**Corollary 4** *After passage of a full schedule of flows, the resulting charge is the marginal charge $\mathcal{K}_f$ of $f$.*

**Proof** After the passage of a full schedule of flows an active flow will, by definition, have passed along every edge of $\mathcal{T}$ hence by Corollary 3 every separator $S \in \mathcal{S}$ will have potential $f_S$. Furthermore every clique $C \in \mathcal{C}$ will be live since it will have received active flows from each of its neighbours. Thus, by Corollary 2, every clique $C$ will have potential $f_C$. Hence the result.

□

**Corollary 5** *Suppose that $f$ factorises on $\mathcal{T}$. Then $\mathcal{K}_f$ is a representation for $f$, and thus:*

$$f = \frac{\prod_{C \in \mathcal{C}} f_C}{\prod_{S \in \mathcal{S}} f_S} \tag{2.9}$$

**Proof** If $f$ factorises on $\mathcal{T}$ then, by definition, there exists a representation $\mathcal{K} = (\{a_C : C \in \mathcal{C}\}, \{b_S : S \in \mathcal{S}\})$ for $f$ on $\mathcal{T}$. Now consider passing a full schedule of flows as defined by Equation 2.5 then by Corollary 4 the resulting charge is the marginal charge $\mathcal{K}_f$ of $f$. But, by Lemma 4, since the passage of a flow does not affect the contraction of a charge then $\mathcal{K}_f$ is a representation for $f$. But since $\mathcal{K}_f$ is a representation for $f$ then $f$ factorises on $\mathcal{T}$ and hence Equation 2.9 holds.

□

**Definition 25** *Let $C'$ and $C^*$ be two cliques of a junction tree $\mathcal{T}$ with potentials $a_{C'}$ and $a_{C^*}$ respectively. Suppose that $S$, the separator joining $C'$ and $C^*$, has potential $b_S$. Then $C'$ and $C^*$ are said to be* **consistent** *if:*

$$\sum_{C' \setminus S} a_{C'} \propto b_S \propto \sum_{C^* \setminus S} a_{C^*}$$

The tree of cliques is said to be *locally consistent* if all mutual neighbours in the tree are consistent. A stronger form of consistency is that of *calibration*.

**Definition 26** *Let $C'$ and $C^*$ be two cliques of a junction tree $\mathcal{T}$ with potentials $a_{C'}$ and $a_{C^*}$ respectively. Suppose that $S$, the separator joining $C'$ and $C^*$, has potential $b_S$. Then $C'$ and $C^*$ are said to be* **calibrated** *if:*

$$\sum_{C' \setminus S} a_{C'} = b_S = \sum_{C^* \setminus S} a_{C^*}$$

**Theorem 15** *If a flow is passed across a separator which joins two mutually calibrated cliques then the potentials of the cliques and separator are invariant to the flow.*

**Proof** Let $S$ be a separator joining two mutually calibrated cliques $C'$ and $C^*$. Let $b_S$, $a_{C'}$ and $a_{C^*}$ denote the potentials associated with $S$, $C'$ and $C^*$ respectively. Without loss of generality let us consider the passage of a flow from $C'$ to $C^*$ then by definition the potential on $a_{C'}$ is unchanged. The new potential $b_S^*$ on $S$ is:

$$b_S^* = \sum_{C' \setminus S} a_{C'} = b_S$$

and the update factor $\lambda_S$ equals one so $a_{C^*}$ is unaffected by the passage of the flow also.

□

When a full schedule of flows has been passed, every pair of neighbouring cliques will be calibrated and hence no additional flow will affect the potentials comprising the charge. The system is thus said to have reached *equilibrium*.

### 2.12.2 Evidence Entry

We will now consider how we may incorporate evidence into a probabilistic expert system. The exact methodology will depend upon the way in which the functions concerned are being modelled. This is a subject will be considered in more depth

in latter chapters. We will, however, introduce some of the basic concepts here. We will consider how evidence may be entered on a variable $X_a$ which is modelled through the use of algebraic formulae which are functions of a symbolic variable $x_a$. A continuous variable defined by its probability density function may be represented in this way. Similarly a discrete random variable which can not be partitioned into a finite range of possible states may be modelled similarly (for example, a Poisson distribution might be treated in this way). We shall say that such variables have been represented *symbolically*. The second type of functional representation we shall consider is that which may be applied to a discrete random variable which may be constrained to a finite range of possible states (for example a multinomial distribution or a truncated Poisson). In this case the functions on the random variables are represented by *potential tables* where each cell of the table holds a function value for a particular realisation of the random variables which have been constrained to finite states. Each function value may be a function of the variables in that potential which have been represented symbolically.

Let $f = f_K$ be the joint density describing the initial uncertainty about the variables $X = (X_1, X_2, \ldots, X_k)$ in the system. Suppose that new evidence of the form $\mathcal{E} : X_a = x_a^*$ is observed for all $a \in K^* \subseteq K$, then we will wish to update our uncertainty appropriately. This will require us to calculate the joint conditional density of $X$ given $\mathcal{E}$. Let $\mathcal{E}_a$ denote the partial evidence that $X_a = x_a^*$ for $a \in K^* \subseteq K$ and let $e_a$ be some function which enters the evidence $\mathcal{E}_a$.

Suppose that $X_a$ is a random variable which is symbolically represented then the corresponding function $e_a$ will need to replace every $x_a$ in the system with $x_a^*$. Consider some potential function $\phi(x_{K'})$ for which $x_{K'} = \{x_{K' \setminus a}, x_a\}$ then we may write $\phi(x_{K'})$ as $\phi(x_{K' \setminus a}, x_a)$. The *evidence function* $e_a$ is thus defined to be:

$$e_a(\phi(x_{K' \setminus a}, x_a)) = \phi(x_{K' \setminus a}, x_a^*)$$

Assuming that $\phi(x_{K'}) = f_{K'}$, the joint density of the random variables $X_{K'}$, then $e_a(\phi(x_{K'})) = f_{X_{K'} | X_a = x_a^*} \times f_{X_a = x_a^*}$ where $f_{X_a = x_a^*}$ is a constant. We may determine $f_{X_a = x_a^*}$ by marginalising $e_a(\phi(x_{K'}))$ over the random variables $X_{K' \setminus a}$. Thus the required conditional density of $X_{K' \setminus a} \mid \mathcal{E}_a$ may be calculated as follows:

$$f_{X_{K'} | X_a = x_a^*} = \frac{e_a(f_{K'})}{\sum_{K' \setminus a} e_a(f_{K'})}$$

$f_{X_a = x_a^*}$ is termed a *normalisation constant*.

Thus our strategy for the entry of evidence on symbolically represented variables into the system must first involve the replacement of every $x_a$ with $x_a^*$ for $a \in K^* \subseteq K$ using the functions $e_a$. This must take place in every clique and separator potential which is a function of at least one $x_a$ for $a \in K^*$.

Now consider how we may enter evidence on a random variable $X_a$ which is represented by a potential table. Let $e_a$ be an *evidence vector* defined on $\mathcal{X}_a$ such that $e_a(x_a) = 1$ if $x_a = x_a^*$, and 0 otherwise. Consider multiplying $f_K$ by $e_a$ the resulting function $f_K^*$, say, will be zero for $x_a \neq x_a^*$ and for $x_a = x_a^*$ be equal to the product of $f_{X_{K\setminus a}|X_a=x_a^*}$ and $f_{X_a=x_a^*} = P(X_a = x_a^*)$. Hence, in order to enter evidence on a variable $X_a$ represented by a potential table we need only multiply one of the clique potentials which is a function of $x_a$ by the evidence vector $e_a$. We may normalise the system to obtain the function $f_{X_{K\setminus a}|X_a=x_a^*}$ by marginalising to determine the normalisation constant $P(X_a = x_a^*)$ and then dividing by it.

Now suppose that the evidence $\mathcal{E}$ has been entered into the system using either evidence functions, evidence vectors or a combination of the two then the derived function $f_{X\&\mathcal{E}}$ is not a density (since it does not marginalise to one) and the cliques are not necessarily mutually calibrated. We must therefore pass a propagation schedule through the tree in order to calibrate the cliques. Marginalising any clique or separator with respect to all the random variables it contains we are able to determine the normalisation constant and hence divide every clique and separator potential by it. The charge on $\mathcal{T}$ is now a representation for $f_{X|\mathcal{E}}$ as required and the potential function on any clique $C \in \mathcal{C}$ (or separator $S \in \mathcal{S}$ ) is the margin of $f_{X|\mathcal{E}}$ on $C$ (or $S$ ).

It should be noted that if the evidence $\mathcal{E}$ consists of purely discrete evidence then the normalisation constant represents the probability $P(\mathcal{E})$ of that collection of evidence. If continuous evidence is being added then the normalisation constant is less informative.

## 2.13 Discussion

Causal probabilistic networks have a long history in statistics and decision making. It was not until the early 1980s, however, that they were applied to the field of expert systems by Pearl (1982). Propagation algorithms which operate on causal probabilistic networks may be found in Pearl (1988). Such algorithms are primarily

restricted to work on singly connected networks, although Pearl has made several extensions to his scheme.

In this chapter we have concentrated on the representation of probabilistic expert systems by junction trees. The purpose of this is that junction trees allow the application of the propagation algorithm devised by Lauritzen & Spiegelhalter (1988). This method, by converting the problem to an undirected tree structure, avoids the problems associated with nonsingly connected networks.

# Chapter 3

# Numeric Techniques

## 3.1 Introduction

This chapter seeks to introduce numeric methodologies which enable the construction of PESs. Two classes of PES are described. The first is the archetypical PES termed the *discrete exact case* (see Dawid *et al.*, 1993). In this model random variables are assumed to take discrete distributions which possess a finite number of states. The joint probability distribution of $n$ such variables may then be expressed as an $n-$dimensional probability table. Each cell in the table corresponds to a particular combination of states of the random variables. The entry in each cell is a pure number representing the exact probability of obtaining that particular combination of states. We show how a rule-based expert system may be thought of as a special form of the discrete exact case. A standard model composed of bernoulli random variables is presented to illustrate the discrete exact case. We consider how such models may be programmed using a numeric language such as *Dyalog APL* or one which possesses numeric features such as *Mathematica*, and explore the range of functions that will be required. Two extensions of the propagation scheme are also described. One technique termed *maximisation* identifies the most likely realisation(s) of the joint state space. The second technique, termed *simulation*, provides random realisations of the joint state space which have been generated in a way which is consistent with the underlying independence network.

It should be noted that this class of models is not the only "discrete exact case" if the term is to be taken literally. In fact other models composed solely of discrete variables which are manipulated using exact techniques may also be constructed. These do, however, employ the use of exact formulae which will, in an automated system, require the use computer algebra rather than computer

arithmetic. Their key distinction from the discrete exact case is either that they model discrete variables with a non-finite state space or that they comprise some unknown parameter which has to be represented symbolically. Examples of such models will be presented in Chapter 4.

The second class of PES we investigate in this chapter is a mixed model composed of both discrete and continuous variables. The discrete variables are assumed to possess a finite number of states and have no continuous parents. The continuous variables are conditionally normal given their parents which may be discrete and/or continuous. We present Lauritzen's scheme (Lauritzen, 1992) for dealing with this model. It too is an exact methodology, however, it does not allow the full information available on the continuous random variables to be represented. Instead it only enables the propagation of the probabilities, means and variances of the underlying variables. A standard example, Lauritzen's "waste incinerator problem", is presented to illustrate this scheme. This example seeks to model the processes occurring in a waste incinerator plant. We show how an expert system based on Lauritzen's scheme may perhaps be an oversimplification of this model. We extend Lauritzen's scheme to provide a methodology for simulation in mixed graphical association models and show how this may be used to either overcome or identify some of the limitations of the scheme.

## 3.2 Discrete Exact Case

Suppose we are given an independence graph $\mathcal{G} = (K, E)$ for a set of random variables $X = (X_1, X_2, \ldots, X_k)$ such that each vertex $a \in K$ corresponds to a random variable $X_a$. Further suppose that no random variable is dependent upon an unknown parameter. Then if, for every $a \in K$, $X_a$ takes values $x_a$ in a finite discrete space $\mathcal{X}_a$ the PES thus defined is an example of the *discrete exact case*. The marginal density of each $X_a$ takes the form:

$$P(X_a = x_a) = p_a(x_a) \quad \text{for} \quad x_a \in \mathcal{X}_a$$

where $p_a(x_a)$ is a real number and $\sum_{x_a \in \mathcal{X}_a} p_a(x_a) = 1$. Similarly the conditional probability distribution, $X_a \mid X_{pa(a)}$, of any random variable $X_a$ given its parents, $X_{pa(a)}$, may be defined as:

$$P(X_a = x_a \mid X_{pa(a)} = x_{pa(a)}) = p_a(x_a; x_{pa(a)}) \quad \text{for } x_a \in \mathcal{X}_a \text{ and } x_{pa(a)} \in \mathcal{X}_{pa(a)} \tag{3.1}$$

where $p_a(x_a; x_{pa(a)})$ is a real number and $\sum_{x_a \in \mathcal{X}_a} p_a(x_a; x_{pa(a)}) = 1$.

The discrete exact case is the archetype of a PES and the simplest form which may be devised. Like a rule-based expert system it enables the storage and application of categorical knowledge by virtue of the fact that all the random variables are discrete. The conditional distribution of any random variable $X_a$ given its parents $X_{pa(a)}$ as defined by Equation 3.1 may be represented by a table of probabilities. Each dimension in the table corresponds to exactly one of the set of variables $\{X_a, X_{pa(a)}\}$. The number of dimensions in the table is thus one plus the number of parents of $X_a$. Every *cell* in the table corresponds to a different realisation of the space $\mathcal{X}_{(a,pa(a))}$. A table of probabilities is the most natural way in which a probability distribution may be represented in the discrete exact case.

## 3.3 The Chest Clinic Example

In this section we will define the chest clinic example which was first introduced by Lauritzen & Spiegelhalter (1988). This simple problem seeks to model a small piece of fictitious medical 'knowledge':

> Shortness-of-breath (dyspnoea) may be due to tuberculosis, lung cancer or bronchitis, or none of them, or more than one of them. A recent visit to Asia increases the chances of tuberculosis, while smoking is known to be a risk factor for both lung cancer and bronchitis. The results of a single X-ray do not discriminate between lung cancer and tuberculosis, as neither does the presence or absence of dyspnoea.

We will use this example to illustrate how the discrete exact case may be programmed using a computer language. In particular the way in which potential functions for the discrete exact case may be represented, and how the computations required by a propagation scheme may be constructed.

In line with Lauritzen & Spiegelhalter a random variable and its corresponding node are both denoted by a single lower case Greek letter. The equivalent lower case Latin letter is used to denote a positive response to the random variable, while a negative response is denoted by overlining this lower case Latin letter. Eight binary variables are used to illustrate the problem. These comprise: $\alpha$ "visit to Asia?", $\beta$ "bronchitis?", $\delta$ "dyspnoea?", $\epsilon$ "either tuberculosis or lung cancer?", $\lambda$ "lung cancer?", $\sigma$ "smoking?", $\tau$ "tuberculosis?", and $\xi$ "positive X-ray?". The

directed acyclic graph describing the scheme is given in Figure 3.1. Further details as to its construction may be found in Lauritzen & Spiegelhalter. It should be noted that the binary variable $\epsilon$ is a logical variable rather than a random variable.

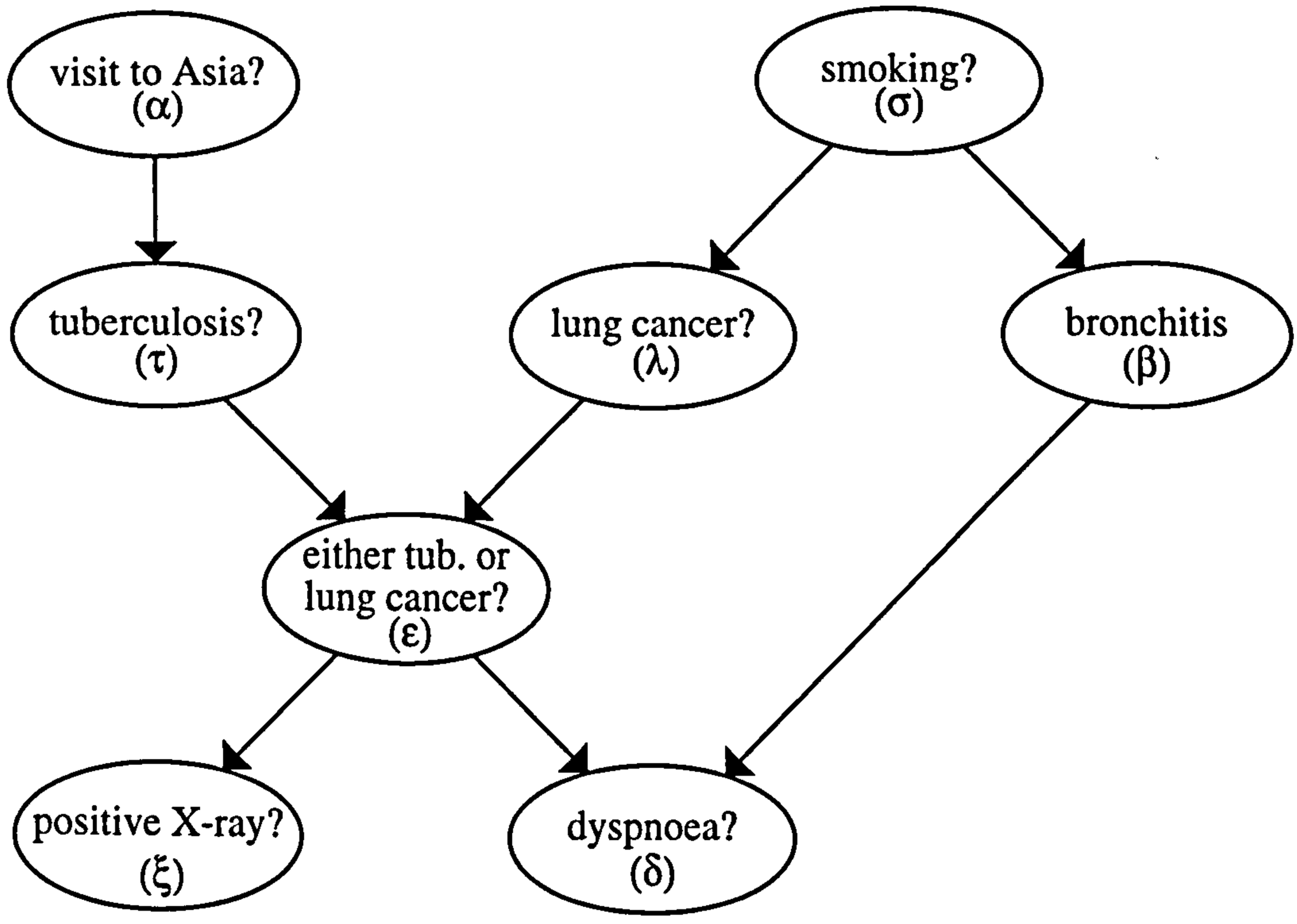



Figure 3.1: The directed acyclic graph illustrating the chest clinic example.

The junction tree formed from the directed acyclic graph in Figure 3.1 is given in Figure 3.2. The set of cliques $\mathcal{C} = \{\{\alpha, \tau\}, \{\beta, \delta, \epsilon\}, \{\beta, \epsilon, \lambda\}, \{\beta, \lambda, \sigma\}, \{\epsilon, \lambda, \tau\}, \{\epsilon, \xi\}\}$ are shown as ovals. The set of separators $\mathcal{S} = \{\{\beta, \epsilon\}, \{\beta, \lambda\}, \{\epsilon\}, \{\epsilon, \lambda\}, \{\tau\}\}$ are shown as squares.

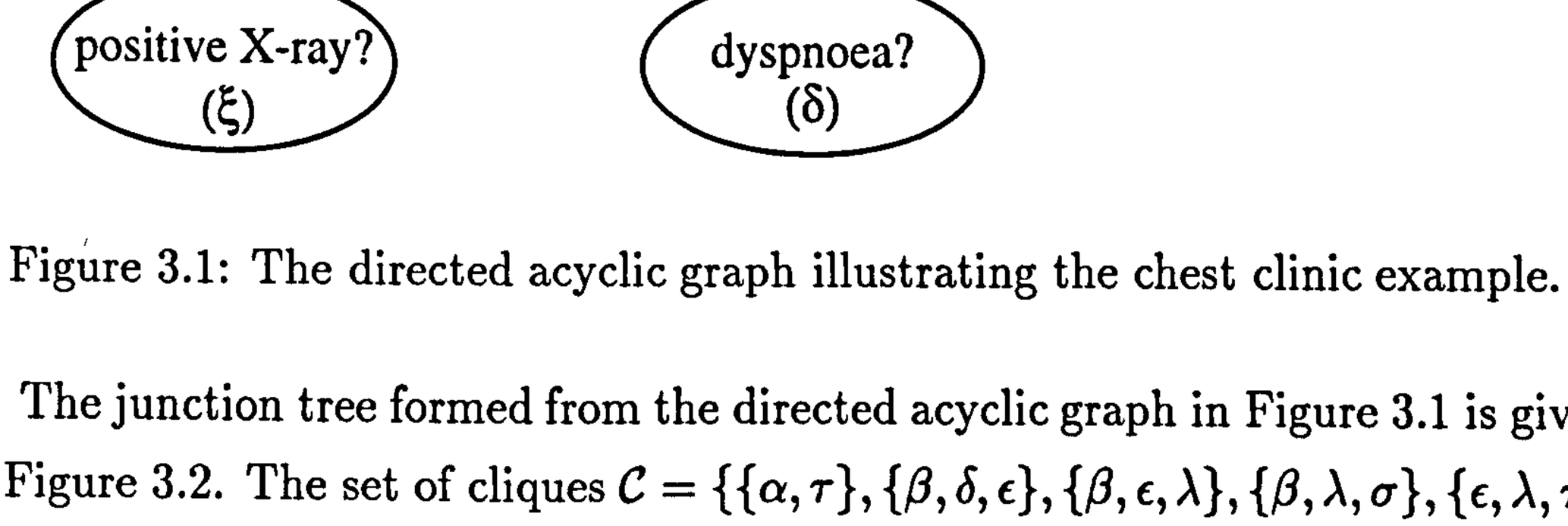



Figure 3.2: A junction tree for the chest clinic example.

$\alpha$ :

| | $\overline{a}$ | $a$ |
|---|---|---|
| $\alpha$ | 0.99 | 0.01 |

$\beta \mid \sigma$ :

| | $\beta$ | |
|---|---|---|
| $\sigma$ | $\overline{b}$ | $b$ |
| $\overline{s}$ | 0.70 | 0.30 |
| $s$ | 0.40 | 0.60 |

$\delta \mid \beta, \epsilon$ :

| | | $\delta$ | |
|---|---|---|---|
| $\beta$ | $\epsilon$ | $\overline{d}$ | $d$ |
| $\overline{b}$ | $\overline{e}$ | 0.90 | 0.10 |
| $\overline{b}$ | $e$ | 0.30 | 0.70 |
| $b$ | $\overline{e}$ | 0.20 | 0.80 |
| $b$ | $e$ | 0.10 | 0.90 |

$\epsilon \mid \lambda, \tau$ :

| | | $\epsilon$ | |
|---|---|---|---|
| $\lambda$ | $\tau$ | $\overline{e}$ | $e$ |
| $\overline{l}$ | $\overline{t}$ | 1 | 0 |
| $\overline{l}$ | $t$ | 0 | 1 |
| $l$ | $\overline{t}$ | 0 | 1 |
| $l$ | $t$ | 0 | 1 |

$\lambda \mid \sigma$ :

| | $\lambda$ | |
|---|---|---|
| $\sigma$ | $\overline{l}$ | $l$ |
| $\overline{s}$ | 0.99 | 0.01 |
| $s$ | 0.90 | 0.10 |

$\sigma$ :

| | $\overline{s}$ | $s$ |
|---|---|---|
| $\sigma$ | 0.50 | 0.50 |

$\tau \mid \alpha$ :

| | $\tau$ | |
|---|---|---|
| $\alpha$ | $\overline{t}$ | $t$ |
| $\overline{a}$ | 0.99 | 0.01 |
| $a$ | 0.95 | 0.05 |

$\xi \mid \epsilon$ :

| | $\xi$ | |
|---|---|---|
| $\epsilon$ | $\overline{x}$ | $x$ |
| $\overline{e}$ | 0.95 | 0.05 |
| $e$ | 0.02 | 0.98 |

Table 3.1: Marginal and conditional probability tables for the chest clinic example.

The marginal and conditional probability distributions for the example are presented as probability tables in Table 3.1. We will let $p(\alpha)$ denote $P(\alpha = a)$. The probability density $f_{\alpha\beta\delta\epsilon\lambda\sigma\tau\xi}$ of the variables $\{\alpha, \beta, \delta, \epsilon, \lambda, \sigma, \tau, \xi\}$ may, by virtue of the factorisation criterion, be written in the form:

$$\begin{aligned} f_{\alpha\beta\delta\epsilon\lambda\sigma\tau\xi} &= p(\alpha\beta\delta\epsilon\lambda\sigma\tau\xi) \\ &= p(\alpha)p(\beta \mid \sigma)p(\delta \mid \beta, \epsilon)p(\epsilon \mid \lambda, \tau)p(\lambda \mid \sigma)p(\sigma)p(\tau \mid \alpha)p(\xi \mid \epsilon) \end{aligned}$$

In general we may arbitrarily assign the initial distribution of a variable to a potential function $a_C$ on one of the cliques $C \in \mathcal{C}$ providing that the clique contains all the variables necessary to define the distribution. In the chest clinic example there is only one such assignment. It is as follows:

$$
\begin{aligned}
a_{\alpha\tau}(\alpha\tau) &= p(\alpha)p(\tau \mid \alpha) \\
a_{\epsilon\xi}(\epsilon\xi) &= p(\xi \mid \epsilon) \\
a_{\epsilon\lambda\tau}(\epsilon\lambda\tau) &= p(\epsilon \mid \lambda, \tau) \\
a_{\beta\epsilon\lambda}(\beta\epsilon\lambda) &= 1 \\
a_{\beta\lambda\sigma}(\beta\lambda\sigma) &= p(\beta \mid \sigma)p(\lambda \mid \sigma)p(\sigma) \\
a_{\beta\delta\epsilon}(\beta\delta\epsilon) &= p(\delta \mid \beta, \epsilon)
\end{aligned}
$$

## 3.4 Programming Probabilistic Expert Systems

We shall now turn to the problem of how a probabilistic expert system may be represented in a computer program. Several different languages were investigated as to their suitability for this task. The most important ingredient which a suitable programming language must possess is an appropriate data structure which will enable the representation of a multivariate distribution. Code complexity may be reduced and interpretabilty improved by a good choice of data structure. The chest clinic example was successfully programmed in three languages: *Dyalog APL*, *C*, and *Mathematica*. These three languages are very different in structure, each having its own strengths and weaknesses as far as the PES builder is concerned. All three are, however, function based languages - a feature which can improve the structure of the underlying code.

Dyalog APL is a mathematical programming language similar in structure to *APL* (*A Programming Language*) but with additional features. Notable amongst these is its inclusion of a tensor data structure. It is a very high-level numeric language characterised by its use of symbols to denote powerful commands. This results in very compact code which is interpreted at run-time rather than compiled. The lack of a compilation feature and shortcomings in the front-end department make Dyalog APL only really useful for *ad hoc* research projects rather than stand-alone applications. The high-level nature of Dyalog APL may make it inappropriate for the implementation of symbolic techniques although string-manipulation features do exist.

*C* may be termed a programmer's programming language. It is suitable for both systems and applications programming. It is not as high-level as either Dyalog APL or *Mathematica* but contains all the basic building blocks necessary to build complex functions - a feature which may often be more desirable. One of the most useful characteristics of *C* is its rich choice of data structures. These may be customised by combining together different elements into one data structure. Thus, while *C* does not explicitly possess a tensor structure it may be programmed such that it interprets an array as one. While *C* is relatively easy to program to perform numeric calculations it is harder, though not impossible, to program it to perform symbolic manipulations. *C* is, afterall, the chosen language of many a compiler builder. Both *Lex* (*a lexical analyser generator*, Lesk, 1975) and *Yacc* (*yet another compiler-compiler* Johnson, 1975) are written in *C* and may be used to write routines accessible by *C* which will perform the sort of symbolic manipulations which we will require. *C* code must be compiled before it is run. This results in the generation of stand-alone executable code.

*Mathematica* is a general computer system and language intended for mathematical and other applications (Wolfram, 1991). Although it is all too often viewed as just a powerful calculator useful for interactive work, it is actually a flexible programming language with three key capabilities. It performs numerical, symbolic and graphical computations. *Mathematica* is a very high-level programming language. While most traditional programming languages may have up to 30 mathematical functions built in, *Mathematica* boasts over 750. It is thus very useful as a research tool. *Mathematica* is quite limited in terms of the data structures it provides, however, the basic data structure in *Mathematica* is a list which is in fact the only data structure which we actually require. *Mathematica* is one of the few computer languages which will perform symbolic computations without additional programming (see also *MACSYMA*, *REDUCE*, and *muMATH*). Individual data elements may thus be either numbers or equations. *Mathematica* will handle these elements in a unified way, thus numeric and symbolic elements may be included in the same structure and mathematical functions may be applied to elements of either type. *Mathematica*'s symbolic capabilities coupled with its powerful graphical routines make it the ideal environment in which to program PESs comprising continuous random variables. The main drawbacks to *Mathematica* are that it is relatively slow since functions are interpreted at run-time rather than being compiled and it can not be used to build stand alone applications.

In view of the comments made above it was decided that *Mathematica* possessed the best combination of features for our purposes. Most of the code written for the examples in this thesis have been programmed in it. *Mathematica* may be seen as the best research language of the three due to its symbolic, graphical and interactive features. Of the three, however, *C* is probably the best choice for the creation of a stand-alone PES implementing our techniques.

## 3.5 Data Structures

We shall briefly identify possible data structures which may be used to represent a discrete exact PES in each of the three main programming languages considered - Dyalog APL, *C*, and *Mathematica*. The data structures required fall into two distinct categories. The first requirement is a structure which may be used to represent a potential table for either a clique or a separator. The second requirement is a structure which may be used to represent a junction tree itself. It should be noted that simpler structures would first be needed if one wishes to perform the *compilation phase*, in which an initial graph is moralised and triangulated to form a junction tree, and the *initialisation phase*, in which the initial clique potentials are formed. These details will be neglected at this time however.

Dyalog APL has a much richer data structure than APL which provides a superior environment for the PES builder. Three features of particular interest are: that it allows both strings and numbers to be represented in the same data structure; it allows the representation of tensors in addition to arrays; and tensors composed of elements of different types and dimensions may be constructed. All three of these features will be exploited in our example data structures. There follows a Dyalog APL definition of one of the initial clique potentials ($\beta\lambda\sigma$) in the chest clinic example.

```
POT5 ← (1 3 ρ 'BETA' 'LAMBDA' 'SIGMA')(2 2 2 ρ 0.3465 0.18
       0.0035 0.02 0.1485 0.27 0.0015 0.03)
```

Figure 3.3 gives a graphical representation of the defined structure in which each box corresponds to an array.

The clique potential structure thus defined is an array of two elements. The first element identifies the variables in the clique. The second element contains the

numeric data pertaining to that clique potential, in otherwords it is the potential table. The list of variables is represented by an array each element of which is a character array giving the name of a particular variable. The ordering of the list of variables corresponds to the ordering of the potential table. The potential table is a multi-dimensional array (or *tensor*). It is structured in layers. Each layer corresponds to a variable and is an array with elements which denote the different levels of that variable. The innermost element of all the layers for a given combination of levels of variables is a number corresponding to the potential put on that realisation. The first variable name in the list of variables represents the outermost layer of the potential table. Each proceeding variable name represents the next inner layer of the potential table.

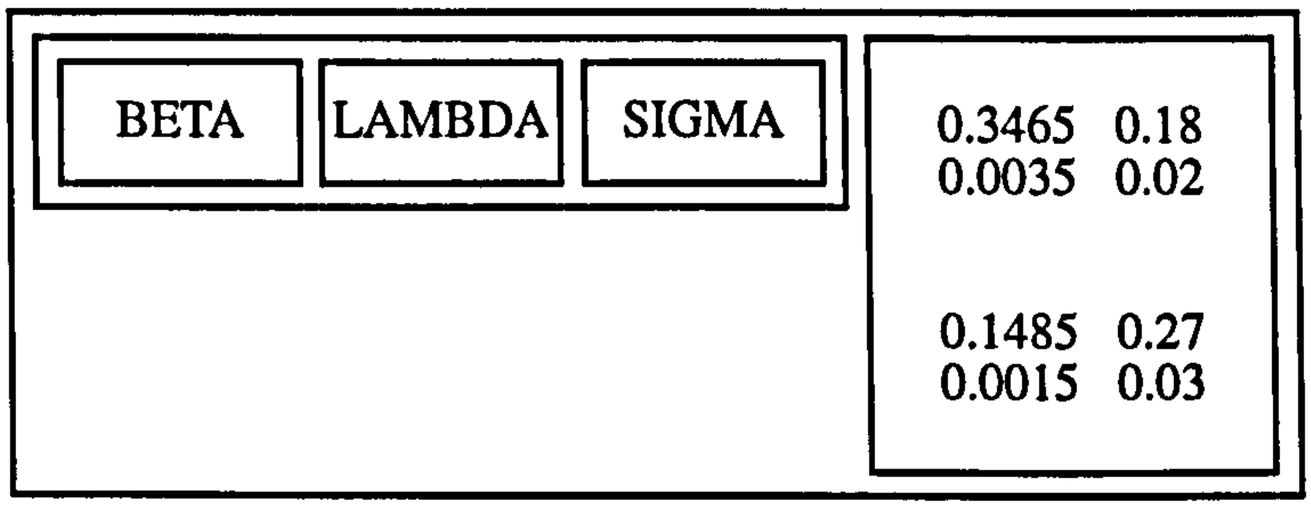



Figure 3.3: A graphical representation of a clique potential.

Separator potentials may be represented in a similar fashion. It should be noted however, that since separators do not necessarily contain a unique combination of variables an additional indicator is required to identify each one. A possible solution to this is to number each separator arbitrarily. Thus a separator for $\beta\lambda$ may be defined as:

```
SEP4 ← ((1 2 ρ 'BETA' 'LAMBDA')(4))(2 2 ρ 1)
```

Figure 3.4 gives a graphical representation of this separator.

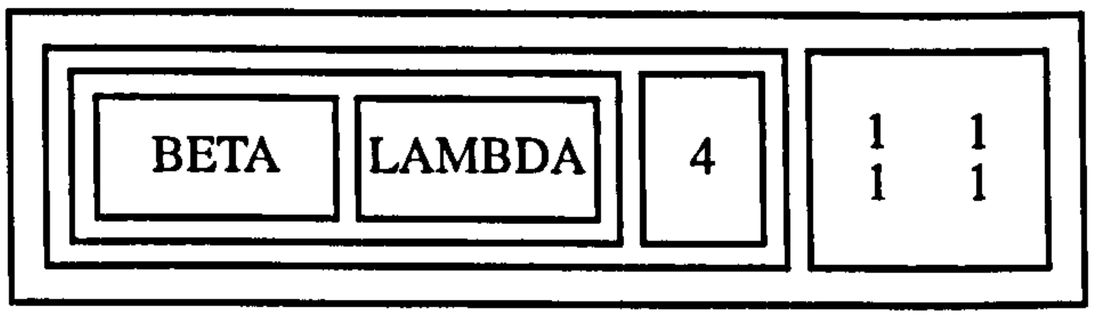



Figure 3.4: A graphical representation of a separator potential.

The structure of a junction tree is explicitly contained in its propagation schedule. If a palindromic propagation schedule is employed then we need only define

the first half of this schedule to define our junction tree structure. Each flow in the schedule may be represented by an array the first element of which identifies the source clique, the third element of which identifies the sink clique and the associated separator is denoted by the second element. The first flow may be defined thus:

```
FLOW1 ← (1 2 ρ 'ALPHA' 'TAU')
        (('TAU')(1))(1 3 ρ 'EPSILON' 'LAMBDA' 'TAU')
```

It is represented in Figure 3.5.

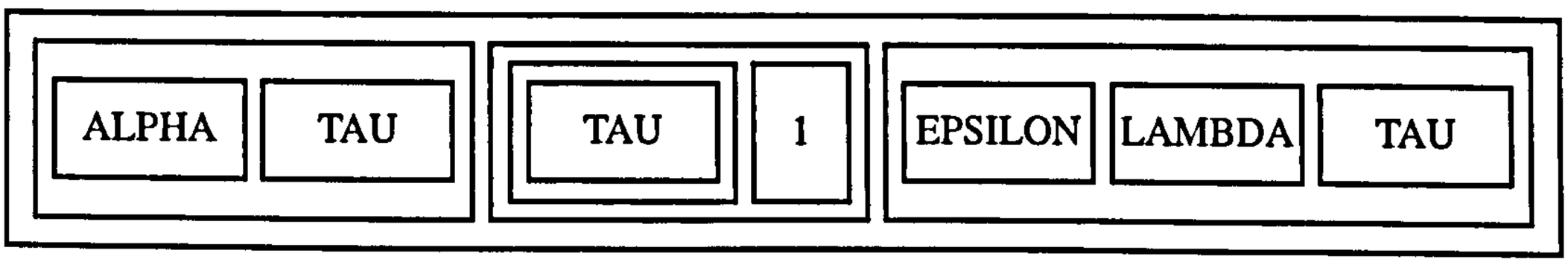



Figure 3.5: A graphical representation of a flow.

*C* not only supports built-in data types but also user-defined data types which allow the programmer to tailor the data representation to the application. A user-defined data type which groups together variables of different data types is known as a structure and is defined by the command `struct`. There follows a suitable definition for two structures which together may be used to define a clique potential:

```
typedef struct variable_name {
        char                name[MAXCHARS];
} VARIABLE_NAME;

typedef struct clique {
        VARIABLE_NAME       var[MAXVARS];
        int                 num;
        int                 level[MAXVARS];
        float               data[MAXDATA];
} CLIQUE;
```

`CLIQUE` is a structure composed of four distinct elements: `var`, `num`, `level`, and `data`. `var` is an array of character arrays. Each component character array, `name`,

say, holds the name of a variable with maximum length MAXCHARS. MAXVARS is the maximum number of variables allowable in a clique. The order of the variables in var determines the order by which the other data structures are interpreted. num is an integer holding the number of variables in the clique. level is an integer array of length MAXVARS the $n-$th element of which gives the number of levels of the $n-$th variable in var. data is an array of floats (a float is a number from approximately $10^{-38}$ to $10^{38}$ in absolute value). It may be used to store the potential table despite the fact that it is flat in structure. For example, suppose a clique contains $N$ variables $X = (X_1, X_2, \ldots X_N)$ and let each variable $X_i$ take levels $0, 1, \ldots, N_i$ for $i = 1, 2, \ldots, N$. The potential corresponding to a realisation $x = (x_1, x_2, \ldots, x_N)$ may be stored at the $p-$th position of an array where:

$$p = x_1 + \sum_{i=2}^{N} x_i \prod_{j=1}^{i-1} N_j$$

and 0 denotes the first position in the array. The following piece of $C$-code defines the initial potential of clique $\beta\lambda\sigma$:

```
main(){
  CLIQUE bls;
  strcpy(bls.var[1], ''BETA'');
  strcpy(bls.var[2], ''LAMBDA'');
  strcpy(bls.var[3], ''SIGMA'');
  bls.num = 3;
  bls.level[1] = 2;
  bls.level[2] = 2;
  bls.level[3] = 2;
  bls.data[1] = 0.3465;          /* (b=0, l=0, s=0) */
  bls.data[2] = 0.1485;          /* (b=1, l=0, s=0) */
  bls.data[3] = 0.0035;          /* (b=0, l=1, s=0) */
  bls.data[4] = 0.0015;          /* (b=1, l=1, s=0) */
  bls.data[5] = 0.18;            /* (b=0, l=0, s=1) */
  bls.data[6] = 0.27;            /* (b=1, l=0, s=1) */
  bls.data[7] = 0.02;            /* (b=0, l=1, s=1) */
  bls.data[8] = 0.03;            /* (b=1, l=1, s=1) */
}
```

Separators may be defined in a similar way to cliques using the struct construct. As discussed above an additional variable is required to uniquely identify them however. This is the variable sep_num in the following piece of code:

```
typedef struct separator {
      VARIABLE_NAME  var[MAXVARS];
      int            sep_num;
      int            num;
      int            level[MAXVARS];
      float          data[MAXDATA];
} SEPARATOR;
```

The structure of the junction tree may again be defined by the first half of a palindromic propagation schedule. The following nested definitions give a suitable format for this schedule:

```
typedef struct clique_name {
      VARIABLE_NAME  var[MAXVARS];
} CLIQUE_NAME;

typedef struct sep_name {
      VARIABLE_NAME  var[MAXVARS];
      int            sep_num;
} SEP_NAME;

typedef struct flow {
      CLIQUE_NAME    source;
      SEP_NAME       sep;
      CLIQUE_NAME    sink;
} FLOW;

typedef struct sched {
      FLOW           schedule[NUMSEPS];
} SCHED;
```

SCHED is the structure of the schedule. It consists of an array, `schedule`, of individual flows of type `FLOW`. There are `NUMSEPS` such flows, where `NUMSEPS` is defined to be the number of separators in the junction tree. A flow is defined by the names of the three universes associated with it - `source`, the source clique, `sink`, the sink clique, and `sep`, the associated separator. `source` and `sink` are both of type `CLIQUE_NAME` which is an array, `var`, of variable names each of type `VARIABLE_NAME`. `sep` is of type `SEP_NAME` which consists of an array, `var`, of variable

names each of type `VARIABLE_NAME`, and a unique integer identifier `sep_num`.

The most flexible and powerful object in *Mathematica* is a list. Lists may be used to collect together several expressions of any kind. Vectors, matrices, and tensors may all be represented by lists, the former, trivially, as a list, the latter two by nesting lists within lists. We may define the initial clique potential for $\beta\lambda\sigma$ in *Mathematica* as follows:

```
bls = {{ beta, lambda, sigma }, {{{ 0.3465, 0.18 }, { 0.0035, 0.02 }},
       {{ 0.1485, 0.27 }, { 0.0015, 0.03 }}}}
```

The clique potential `bls` is defined to be a list of two elements. The first element is a list of variable names the order of which determines the order in which the potential table is stored - outermost layer first. The second element is a tensor of rank 3 which stores the potential table.

The propagation schedule, and hence structure of the junction tree, may also be defined using the list construct. The schedule `sched` (see below) for the chest clinic example is defined as a list of five elements. Each element is a list of three elements - a source clique, a separator, and a sink clique. The two cliques are defined by lists of variable names. The separator is a list of two elements, the former being a list of variable names, the latter an integer uniquely defining that separator.

```
sched =  { { { alpha, tau }, { { tau }, { 1 } },
           { epsilon, lambda, tau } },
           { { epsilon, xi }, { { epsilon }, { 2 } },
           { epsilon, lambda, tau } },
           { { epsilon, lambda, tau }, { { epsilon, lambda }, { 3 } },
           { beta, epsilon, lambda } },
           { { beta, lambda, sigma }, { { beta, lambda }, { 4 } },
           { beta, epsilon, lambda } },
           { { beta, delta, epsilon }, { { beta, epsilon }, { 5 } },
           { beta, epsilon, lambda } } }
```

## 3.6 Functions Required for Propagation

In this section we investigate the functions required to construct a discrete exact PES, pass a propagation schedule and enter evidence into the system. These functions may be broadly classified into three distinct types: *manipulative operations*, *numeric operations*, and *complex operations*. We shall consider each class in turn and present some simple *Mathematica* code in order to demonstrate how each required operation may be constructed. Although *Mathematica* code is presented rather than pseudocode we feel that *Mathematica* is readable enough to suffice. Additional specifics of syntax may be found in Wolfram (1991). The functions described may all be readily adapted to suit other languages providing that the underlying structure of the potentials in the chosen language is observed. In all cases a potential is assumed to have been defined to be a list of two elements - the first element being a list of the variables in the potential, and the second element being the potential table itself.

### 3.6.1 Manipulative Operations

Manipulative operations may be applied to the potential of either a single clique or a single separator. They do not alter the information contained in the potential but rather *manipulate* it into a more desirable format. This new format simplifies the interaction of two potentials. Two manipulative operations are required - an *extension* operator and a *re-ordering* operator. The extension operator extends a potential table to occupy a table of higher rank. The `reorder` function re-orders the potential table to match some desired ordering. A function, `rearrange`, which both extends and re-orders a potential is also presented.

The *Mathematica* function `extend` takes three parameters: `pottable` the original potential table, `addvar` the additional variable, and `numlev` the number of levels of the additional variable. It outputs `newpot` the original potential extended to incorporate `addvar` as an additional outer layer. The `Join[list1, list2]` function is used to join two lists together, and the `Table[expr, i]` function is used to generate a list composed of `i` duplicates of `expr`.

```
extend[pottable_, addvar_, numlev_] := Block[{vars, data,
      newvars, newdata, newpot },
      vars = pottable[[1]];
      data = pottable[[2]];
      newvars = Join[ { addvar }, vars];
      newdata = Table[data,  { numlev }];
      newpot = Join[{newvars}, {newdata}];
      Return[newpot]
]
```

The function `reorder` takes two parameters: `pottable` the original potential table, and `neword` the ordering of the variables required. The local variable `ord` is a list which gives the position of each of the original variables in the list `neword`. The `Transpose` function is a multidimensional transposition operator which transposes the potential table into the same order as `neword`. `reorder` returns the re-ordered potential.

```
reorder[pottable_, neword_] := Block[{vars, data, ord,
      newdata, newpot},
      vars = pottable[[1]];
      data = pottable[[2]];
      ord = {};
      Do[ord=Join[ord,{neword}], {Length[vars]}];
      ord = Flatten[MapThread[Position, {ord, vars }]];
      newdata = Transpose[data, ord];
      newpot = Join[{neword}, {newdata}];
      Return[newpot]
]
```

The two manipulative operators `extend` and `reorder` will need to be applied to any potential before a numeric operator may be applied to it. This will ensure that the potential is of the correct dimension and arrangement for the numeric operator. The function `rearrange` will extend and reorder a potential `pottable` so that it contains the same variables, and is in the same order as `neword`.

```
rearrange[pottable_,neword_] := Block[{vars, newpot,
      addvars, newvar, varpos, numlev},
      vars = pottable[[1]];
      newpot = pottable;
      addvars = Complement[neword, vars];
      Do[newvar = addvars[[1]];
         addvars = Drop[addvars, 1];
         varpos = Position[globvars, newvar][[1,1]];
         numlev = globnums[[varpos]];
         newpot = extend[newpot, newvar, numlev],
         {Length[addvars]}];
      newpot = reorder[newpot,neword];
      Return[newpot]
]
```

We assume that `globvars` and `globnums` are global variables. `globvars` is a list of all the variables in the problem, while `globnums` is a list of the number of levels of these variables and is in the same order as `globvars`. These global variables may be defined for the chest clinic example as follows:

```
globvars = { alpha, beta, delta, epsilon, lambda, sigma, tau, xi }
globnums = { 2, 2, 2, 2, 2, 2, 2, 2 }
```

### 3.6.2 Numeric Operations

Three functions applying numeric operations are required. Two, *multiplication* and *division*, combine two potential tables together. The third, *marginalisation* collapses a potential table over a single variable. All three require the original potential table(s) to have been suitably rearranged first.

`mult` takes two parameters `pot1` and `pot2`. Both are potentials containing potential tables of the same order and dimension. `mult` returns a single potential in which each element of the potential table in `pot1` has been multiplied by the corresponding element in `pot2`.

```
mult[pot1_, pot2_] := Block[{vars, data, newpot},
      vars = pot1[[1]];
      data = pot1[[2]] * pot2[[2]];
      newpot = Join[{vars}, {data}];
      Return[newpot]
]
```

`div` is similar to `mult` except it returns a single potential in which each element of the potential table in `pot1` has been divided by the corresponding element in `pot2`. The *Mathematica* function `Divide` is first altered to correct it such that it obeys the definition of division by a zero.

```
Unprotect[Divide]
Divide[a_, 0] := 0
Protect[Divide]

div[pot1_, pot2_] := Block[{vars, data, newpot},
      vars = pot1[[1]];
      data = Divide[pot1[[2]], pot2[[2]]];
      newpot = Join[{vars}, {data}];
      Return[newpot]
]
```

`marg` is a marginalisation operator taking a single parameter `pottable`. This is a potential which has been reordered such that the variable we which to marginalise over is the first variable in the list of variable names, and hence forms the outermost layer of the potential table structure. Thus, viewing the potential table as a list of potential tables in order to perform the marginalisation we simply need to sum the elements of this list. The `Apply[Plus, data]` command facilitates this.

```
marg[pottable_] := Block[{vars, data, newvars,
      newdata, newpot},
      vars = pottable[[1]];
      data = pottable[[2]];
      newvars = Drop[vars, 1];
      newdata = Apply[Plus, data];
      newpot = Join[{newvars}, {newdata}];
      Return[newpot]
]
```

The joint marginal potential of a list of variables `vars` in a given clique potential `pot` may be determined using the `margd` function. The clique potential `pot` is rearranged such that the list of variables, `mvars`, which are in the `pot` but not in the list `vars` form the outermost layers of the potential table in `pot`. The `marg` function is then repeatedly applied to eliminate these variables from the resulting potential `mpot`. If `vars` is a list consisting of a single variable the marginal potential of that variable is returned by `margd`.

```
margd[pot_, vars_] := Block[{mvars,lvars,mpot,i},
      mvars = Complement[pot[[1]], vars];
      lvars = Join[mvars,vars];
      mpot = rearrange[pot, lvars];
      For[i=1,i<=Length[mvars],i++,
        mpot = marg[mpot];
      ];
      Return[mpot];
]
```

### 3.6.3 Complex Operations

Complex operators may now be constructed from these component parts. A set of problem specific global variables will also be required. We will illustrate the structure of these by defining the variables required to construct the chest clinic example. Three complex operators pertaining to three stages of a PES's usage will be described. These concern the *initialisation*, *propagation*, and *evidence entry* phases.

The initial probability distributions of each random variable in the chest clinic example as defined in Table 3.1 may be defined as potentials as follows:

```
inita = {{alpha},{0.99,0.01}}
initb = {{beta, sigma},{{0.70,0.40},{0.30,0.60}}}
initd = {{delta, beta, epsilon},{{{0.90,0.30},{0.20,0.10}},
                                 {{0.10,0.70},{0.80,0.90}}}}
inite = {{epsilon, lambda, tau},{{{1,0},{0,0}},{{0,1},{1,1}}}}
initl = {{lambda, sigma},{{0.99,0.90},{0.01,0.10}}}
```

```
inits  =  {{sigma}, {0.50,0.50}}
initt  =  {{tau, alpha}, {{0.99,0.95},{0.01,0.05}}}
initx  =  {{xi, epsilon}, {{0.95,0.02}, {0.05,0.98}}}
```

init is a list of all the initial probability distributions. The position of an initial probability distribution in init corresponds directly to the position of the variable in globvars whose initial conditional distribution it represents.

```
init = {inita, initb, initd, inite, initl, inits, initt, initx}
```

cliques and seps are lists of the cliques and separators. Their potentials will be stored in two lists named cpots and spots respectively. The positions of the clique and separator potentials in cpots and spots will correspond directly to the position of the cliques and separators in cliques and seps. Similarly assign gives the position dependant assignment of initial probability distributions to each clique potential.

```
cliques  =  {{alpha, tau}, {epsilon, xi}, {epsilon, lambda, tau},
            {beta, epsilon, lambda}, {beta, lambda, sigma},
            {beta, delta, epsilon}}
   seps  =  {{{tau},{1}}, {{epsilon},{2}}, {{epsilon, lambda},{3}},
            {{beta, lambda},{4}}, {{beta, epsilon},{5}}}
 assign  =  {{alpha, tau}, {xi}, {epsilon}, { },
            {beta, lambda, sigma}, {delta}}
```

The *Mathematica* function initialise is used to initialise the system using the rearrange and mult functions defined earlier. It performs three main operations. Firstly it defines the initial separator potentials spots. Each initial separator potential is a list the first element of which is the list of variables and an integer identifier as defined for that separator in seps. The second element is the potential table for that separator. Every element in the potential table is set equal to one. The second operation performed by initialise is the construction of the initial clique potentials cpots. These are built in a similar way to the initial separator potentials. Lists of variables are obtained from cliques and potential table elements are set equal to one. The final operation performed by initialise is the assignment of the conditional probability distributions of each of the variables

to the respective cliques. Looping over each clique, for every variable assigned to it in `assign`, the corresponding conditional probability distribution in `init` is rearranged and multiplied by the clique's potential as given in `cpots`. That clique potential is then replaced by the resulting product.

```
initialise := Block[{i,cur,data,new,curass,curpot,
j, pos,addpot},
spots = {};
For[i=1,i<=Length[seps],i++,
   cur = seps[[i]];
   data = rearrange[{{},1},cur[[1]]][[2]];
   new = Join[{cur},{data}];
   spots = Join[spots, {new}]
];
cpots = {};
For[i=1,i<=Length[cliques],i++,
   cur = cliques[[i]];
   data = rearrange[{{},1},cur][[2]];
   new = Join[{cur},{data}];
   cpots = Join[cpots, {new}]
];
For[i=1,i<=Length[cliques],i++,
   curass = assign[[i]];
   curpot = cpots[[i]];
   For[j=1,j<=Length[curass],j++,
      cur = curass[[j]];
      pos = Position[globvars, cur][[1, 1]];
      addpot = rearrange[init[[pos]],curpot[[1]]];
      curpot = mult[curpot, addpot]
   ];
   cpots[[i]] = curpot;
];
]
```

The initialised clique potentials, `cpots`, for the chest clinic example are thus:

```
{ {{alpha, tau},{{0.9801,0.0099},{0.0095,0.0005}}},
  {{epsilon, xi},{{0.95,0.05}, {0.02,0.98}}},
  {{epsilon, lambda, tau},{{{1,0}, {0,0}}, {{0,1}, {1,1}}}},
  {{beta, epsilon, lambda},{{{1,1}, {1,1}}, {{1,1}, {1,1}}}},
  {{beta, lambda, sigma},{{{0.3465,0.18}, {0.0035,0.02}},
                          {{0.1485,0.27}, {0.0015,0.03}}}},
  {{beta, delta, epsilon},{{{0.9,0.3}, {0.1,0.7}},
                           {{0.2,0.1}, {0.8,0.9}}}} }
```

Similarly the list of initial separator potentials, `spots`, is:

```
{ { {{tau}, {1}}, {1,1} },
  { {{epsilon}, {2}}, {1,1} },
  { {{epsilon, lambda}, {3}}, {{1,1}, {1,1}} },
  { {{beta, lambda}, {4}}, {{1,1}, {1,1}} },
  { {{beta, epsilon}, {5}}, {{1,1}, {1,1}} } }
```

Following initialisation a propagation schedule must be passed in order to make the potentials on each clique and separator consistent with each other, and to ensure that these potentials are the joint probability distributions of the variables their respective universes contain. A propagation schedule, `sched`, which is appropriate for the chest clinic example was defined in the last section. The function `prop` may be used to pass that schedule. It consists of two sections. The first passes the flows listed in `sched`. The second passes the same list of flows reversed in both order and direction. The local variables `a1`, `a2`, and `b0` are the names of a source clique, a sink clique, and the separator which joins them respectively. The local variables `pa1`, `pa2` and `pb0` determine the positions of these cliques (separator) in `cpots` (`spots`). The `flow` function is called with these three positions.

The `flow` function passes a flow from a source clique $C_1$ to a sink clique $C_2$ via a separator $S_0$. The updated separator potential $b^*_{S_0}$ denoted `bs0star` is calculated by marginalising the source clique potential $a_{C_1}$ over the list of variables `diff` contained in $C_1$ but not in $S_0$, i.e. `diff` is $C_1 \backslash S_0$. The update factor $\lambda_{S_0}$ denoted `l0` is calculated using the `div` function. It is then extended and reordered using the `rearrange` function. The updated potential on the sink clique $a^*_{C_2}$ denoted `a2star` is calculated as the product of its initial potential and the update factor. The two updated potentials are then used to replace their original potentials in `cpots` and `spots`.

```
prop[sched_]:=Block[{i,a1,b0,a2,pa1,pb0,pa2},
      For[i=1,i<=Length[sched],i++,
         a1 = sched[[i,1]];
         b0 = sched[[i,2]];
         a2 = sched[[i,3]];
         pa1 = Position[cliques,a1][[1,1]];
         pb0 = Position[seps,b0][[1,1]];
         pa2 = Position[cliques,a2][[1,1]];
         flow[pa1, pb0, pa2]];
      For[i=Length[sched],i>=1,i--,
         a1 = sched[[i,3]];
         b0 = sched[[i,2]];
         a2 = sched[[i,1]];
         pa1 = Position[cliques,a1][[1,1]];
         pb0 = Position[seps,b0][[1,1]];
         pa2 = Position[cliques,a2][[1,1]];
         flow[pa1, pb0, pa2]];
]
flow[pa1_, pb0_, pa2_] := Block[{b0vars,diff,b0star,
      i,cur,vars,pos,ord,l0,a2star},
      b0vars = spots[[pb0,1]];
      diff = Complement[cpots[[pa1,1]],b0vars[[1]]];
      b0star = cpots[[pa1]];
      For[i=1,i<=Length[diff],i++,
         cur = diff[[i]];
         vars = b0star[[1]];
         pos = Position[vars, cur][[1,1]];
         ord = Join[{cur}, Drop[vars, {pos}]];
         b0star = rearrange[b0star, ord];
         b0star = marg[b0star]];
      l0 = div[b0star, spots[[pb0]]];
      l0 = rearrange[l0, cpots[[pa2,1]]];
      a2star = mult[cpots[[pa2]],l0];
      spots[[pb0]] = Join[{b0vars},{b0star[[2]]}];
      cpots[[pa2]] = a2star;
]
```

The separator potentials `spots` following the passage of a full schedule in the chest clinic example are as follows:

```
{ {{{tau}, {1}},{0.9896,0.0104}},
  {{{epsilon}, {2}},{0.935172,0.064828}},
  {{{epsilon,lambda}, {3}},{{0.935172,0},{0.009828,0.055}}},
  {{{beta,lambda}, {4}},{{0.5265,0.0235}, {0.4185,0.0315}}},
  {{{beta,epsilon}, {5}},{{0.5210244,0.0289756},
                          {0.4141476,0.0358524}}} }
```

Similarly the corresponding clique potentials `cpots` are:

```
{ {{alpha,tau},{{0.9801,0.0099},{0.0095,0.0005}}},
  {{epsilon,xi},{{0.8884134,0.0467586},
                 {0.00129656,0.06353144}}},
  {{epsilon,lambda,tau},{{{0.935172,0},{0,0}},{{0,0.009828},
                         {0.054428,0.000572}}}},
 {{beta,epsilon,lambda},{{{0.5210244,0},{0.0054756,0.0235}},
                         {{0.4141476,0},{0.0043524,0.0315}}}},
  {{beta,lambda,sigma},{{{0.3465,0.18},{0.0035,0.02}},
                        {{0.1485,0.27},{0.0015,0.03}}}},
  {{beta,delta,epsilon},{{{0.46892196,0.00869268},
   {0.05210244,0.02028292}},{{0.08282952,0.00358524},
   {0.33131808,0.03226716}}}}}
```

The marginal distributions of the variables in the system, obtained by use of the `margd` function are as follows:

```
{{alpha}, {0.99, 0.01}}
{{beta}, {0.55, 0.45}}
{{delta}, {0.564029, 0.435971}}
{{epsilon}, {0.935172, 0.064828}}
{{lambda}, {0.945, 0.055}}
{{sigma}, {0.5, 0.5}}
{{tau}, {0.9896, 0.0104}}
{{xi}, {0.88971, 0.11029}}
```

The `enterev` function enters discrete evidence of the form $X_a = x_a^*$ into the system. The variable name, $X_a$, is denoted `var` and the level of the evidence, $x_a^*$, is denoted `level`. An evidence vector `evvec` is constructed. This is a potential on

var with a potential table set equal to one for $X_a = x_a^*$ and zero elsewhere. The location, pos, of the first clique in cliques which contains var is determined and the evidence vector is extended and re-ordered to match this clique. The clique potential of this clique is then replaced by its product with the evidence vector.

```
enterev[var_, level_] := Block[{evvec,pos},
        evvec = rearrange[{{},0},{var}];
        evvec[[2,level+1]] = 1;
        pos = Position[cliques, var][[1, 1]];
        evvec = rearrange[evvec,cliques[[pos]]];
        cpots[[pos]] = mult[cpots[[pos]], evvec];
]
```

In general the propagation schedule required to make the cliques and separators in a junction tree consistent after the entry of evidence will depend upon into which cliques evidence has been entered. In this simple example, to simplify the complexity of the described functions, we shall take the naive approach of passing a full propagation schedule once more. This approach is ensured to be correct but is not, necessarily, the most computationally efficient. Assuming evidence $\mathcal{E}$ has been entered using enterev and a full propagation schedule has been passed using prop the resulting potentials will be consistent but will not be probability distributions. Rather the joint potential function $f$ of the random variables $X$ is the product of the conditional probability distribution of the variables $X$ given the evidence $\mathcal{E}$ and the probability of the evidence $\mathcal{E}$. In order to determine the probability of the evidence, $P(\mathcal{E})$, we may marginalise any potential over the entire set of variables it contains. In order to normalise the system once more we must divide every potential table by this probability. The function norm accomplishes these two tasks. First it arbitrarily marginalises the first clique potential and assigns the resulting probability, $P(\mathcal{E})$, to the global variable const. It then normalises the clique and separator potentials in turn.

```
norm := Block[{pot, vars, i, data},
      pot = cpots[[1]];
      vars = pot[[1]];
      For[i=1,i<=Length[vars],i++,
        pot = marg[pot];
      ];
      const = pot[[2]];
      For[i=1,i<=Length[cliques],i++,
        vars = cpots[[i,1]];
        data = cpots[[i,2]]/const;
        cpots[[i]] = Join[{vars},{data}];
      ];
      For[i=1,i<=Length[seps],i++,
        vars = spots[[i,1]];
        data = spots[[i,2]]/const;
        spots[[i]] = Join[{vars},{data}];
      ];
]
```

If, in the chest clinic example, the evidence that a patient has visited Asia ($\alpha = 1$), and has dyspnoea ($\delta = 1$) is given the `enterev`, `prop`, and `norm` functions may be used to determine that the probability of this evidence is 0.00450137. The normalised clique potentials are then:

```
{ {{alpha,tau},{{0,0},{0.912249,0.087751}}},
  {{epsilon,xi},{{0.776815,0.040885}, {0.003646,0.178654}}},
  {{epsilon,lambda,tau},{{{0.8177,0},{0,0}},
                         {{0,0.0827747},{0.0945489,0.00497626}}}},
  {{beta,epsilon,lambda},{{{0.111116,0},{0.0409375,0.0365444}},
                          {{0.706584,0},{0.0418372,0.0629808}}}},
  {{beta,lambda,sigma},
          {{{0.100069,0.0519841},{0.00544278,0.0311016}},
           {{0.265569,0.482852},{0.00299908,0.0599817}}}},
  {{beta,delta,epsilon}, {{{0,0},{0.111116,0.0774819}},
                          {{0,0},{0.706584,0.104818}}}} }
```

The normalised separator potentials are:

```
{ {{{tau},{1}},{0.912249,0.087751}},
  {{{epsilon},{2}},{0.8177,0.1823}},
  {{{epsilon,lambda},{3}},{{0.8177,0}, {0.0827747,0.0995251}}},
  {{{beta,lambda},{4}},{{0.152054,0.0365444},
                        {0.748421,0.0629808}}},
  {{{beta,epsilon},{5}},{{0.111116,0.0774819},
                         {0.706584,0.104818}}} }
```

Using `margd` the marginal distributions of the variables may be determined to be:

```
{{alpha}, {0, 1}}
{{beta}, {0.188598, 0.811402}}
{{delta}, {0, 1}}
{{epsilon}, {0.8177, 0.1823}}
{{lambda}, {0.900475, 0.0995251}}
{{sigma}, {0.37408, 0.62592}}
{{tau}, {0.912249, 0.087751}}
{{xi}, {0.780461, 0.219539}}
```

## 3.7 Extensions to the Scheme

Two extensions to the methodology of the discrete exact case are presented in Dawid (1992). These are *maximisation* and *simulation*. Maximisation determines the maximum value of $f$ and in the case of a PES seeks to identify the most likely realisation(s) of the joint state space. Simulation provides random realisations of the joint state space which have been generated in a way which is consistent with the underlying independence network. Both are effected by modifications of the basic propagation algorithm. Analogous to maximisation a *minimisation* routine may also be determined.

### 3.7.1 Maximisation

Let $X = (X_1, X_2, \ldots, X_k)$ be a set of discrete random variables whose independence structure is given by a junction tree $\mathcal{T}$. Let $\mathcal{T}$ be composed of a set of cliques $\mathcal{C}$ and a set of separators $\mathcal{S}$ and let $K = \{1, 2, \ldots, k\}$. Suppose that a function $f$, which factorises on $\mathcal{T}$, is specified by a representation $\mathcal{K}_f = (\{a_C : C \in \mathcal{C}\}, \{b_S : S \in \mathcal{S}\})$ then:

$$f = \frac{\prod_{C \in \mathcal{C}} a_C}{\prod_{S \in \mathcal{S}} b_S}$$

We will consider how we may calculate $\hat{f}$ which is defined to be:

$$\hat{f} = max_K f$$

$\hat{f}$ may be calculated by replacing the sum-flow in the basic propagation algorithm described in Section 2.12.1 by a *max-flow* which may be defined as follows:

**Definition 27** *Let $C_1$ and $C_2$ be adjacent cliques in a junction tree $\mathcal{T}$ which are joined by a separator $S_0$. Then a* **max-flow** *passing from $C_1$ (the* **source***) to $C_2$ (the* **sink***) replaces an original charge $\mathcal{K} = (\{a_C : C \in \mathcal{C}\}, \{b_S : S \in \mathcal{S}\})$, by a new charge $\mathcal{K}^* = (\{a^*_C : C \in \mathcal{C}\}, \{b^*_S : S \in \mathcal{S}\})$ , where:*

$$\begin{aligned} b^*_{S_0} &= max_{C_1 \setminus S_0} a_{C_1} \\ a^*_{C_2} &= a_{C_2} \times \lambda_{S_0} \\ \lambda_{S_0} &= \begin{cases} b^*_{S_0}(x_{S_0}) / b_{S_0}(x_{S_0}) & if\, b_{S_0}(x_{S_0}) > 0 \\ 0 & if\, b_{S_0}(x_{S_0}) = 0 \end{cases} \end{aligned} \tag{3.2}$$

*and all other potentials are unaltered.*

The results and considerations of Section 2.12.1 all continue to hold if every sum-flow (or flow) is replaced by a max-flow, every sum-margin (or margin) is replaced by a *max-margin* and every sum-marginal charge (or marginal charge) is replaced by a *max-marginal charge*. We define a max-margin on a subset $A$ of $K$ to be:

$$f_A = \max_{K \setminus A} f$$

When the system has reached equilibrium the potential on every clique $C \in \mathcal{C}$ will be $\hat{f}_C$, the max-margin on $C$. Similarly the potential on every separator $S \in \mathcal{S}$ will be $\hat{f}_S$, the max-margin on $S$. When this is the case the representation for $f$ is termed the max-marginal charge and is denoted $\hat{\mathcal{K}}_f$. Since $\hat{\mathcal{K}}_f$ is a representation for $f$ then the following alternative expression holds for $f$:

$$f = \frac{\prod_{C \in \mathcal{C}} \hat{f}_C}{\prod_{S \in \mathcal{S}} \hat{f}_S}$$

$\hat{f}$ may now be determined from this representation of $f$ since $\hat{f} = \max_A f_A$ for any $A \in \mathcal{C} \cup \mathcal{S}$.

We shall now consider how we may find $\hat{x}$, when this is uniquely determined. We define $\hat{x}$ as follows:

$$\hat{x} = \arg\max_{x \in \mathcal{X}} f(x)$$

For $A \in \mathcal{C} \cup \mathcal{S}$ we notice that $\hat{x}_A$ is:

$$\hat{x}_A = \arg\max_{x_A \in \mathcal{X}_A} \hat{f}_A(x_A)$$

Hence $\hat{x}_A$ may be calculated directly for every $A \in \mathcal{C} \cup \mathcal{S}$ thus determining $\hat{x}$.

Since $\hat{x}$ need not necessarily be unique we apply a different strategy which may be incorporated into the distribute phase of a max-flow propagation algorithm or applied directly afterwards. Let $C^*$ be some root clique with potential $\hat{f}_{C^*}$ and let $\xi_{C^*}$ be a suitable realisation of $x_{C^*}$ which maximises $\hat{f}_{C^*}$ over $\mathcal{X}_{C^*}$. We may proceed, where necessary, to pass max-flows outwards from $C^*$ following the distribute phase schedule. Suppose the first flow is along a separator $S^*$ to a clique $C'$ then we obtain both $\hat{f}_{S^*}$ and $\hat{f}_{C'}$. We may define $\xi_{S^*} \in \mathcal{X}_{S^*}$ to be the co-ordinates of the variables in $\xi_{C^*}$ which are also in $S^*$. We may determine a value $\xi_{C'}$ for $C'$ which maximises $\hat{f}_{C'}$ over $\mathcal{X}_{C'}$ such that its co-ordinates for the variables in $S^*$ are given by $\xi_{S^*}$. Proceeding in this way we obtain a collection $\{\xi_A : A \in \mathcal{C} \cup \mathcal{S}\}$ for which $\xi_A \in \mathcal{X}_A$. The junction tree property ensures that there exists a $\hat{x} \in \mathcal{X}$ such that $\hat{x}_A = \xi_a$ and $\max_K f$ is achieved at $\hat{x}$.

If $\hat{x}$ is not unique then choices of some $\xi_C$ for $C \in \mathcal{C}$ will have be made. Exploring all such choices will determine the possible values of $\hat{x}$. An appropriate strategy is at each stage of the schedule, for a max-flow from a source clique $C_1$ to a sink clique $C_2$ via a separator $S_0$, determine the list of possible $\xi_{S_o}$ given $\xi_{C_1}$ and the list of possible $\xi_{C_2}$ given $\xi_{S_o}$. The list of possible $\hat{x}$ may be reassembled from these $\xi_A$ for $A \in \mathcal{C} \cup \mathcal{S}$. Alternatively we may divide every clique and separator potential by $\hat{f}$. The positions of the ones in the potential tables will then identify the most probable configuration of all the variables.

If $f$ is the joint density of a set of variables $X$ in a PES then a value of $\hat{x}$ represents the most probable joint configuration of the collection of variables. If a collection of evidence $\mathcal{E}$ is entered into the system then the function $f$ becomes $f^*$ where:

$$f^*(x) = P(X = x \mid \mathcal{E})P(\mathcal{E})$$

The most probable configuration(s) of $X$ given the evidence may be determined by applying our maximisation strategy to the system following the incorporation of evidence. Such configurations may be interpreted as the "best explanation" of the evidence. The most probable configuration of a subgroup of variables $X_A$ having marginalised over the other variables $X \backslash X_A$ may also be determined. This is easy if $A$ is the base of some subtree $\mathcal{T}^*$ of $\mathcal{T}$. The required configuration may then be determined by the use of a hybrid algorithm. First sum-flows need to be passed into $\mathcal{T}^*$ such that the resulting charge for $f$ when restricted to $\mathcal{T}^*$ is the joint density of $X_A$. Max-flows may then be passed within $\mathcal{T}^*$ such that the max-marginal charge on $\mathcal{T}^*$ of the joint density for $X_A$ is obtained and the required configuration(s) may be calculated.

If $f$ is zero then the minimum value of $f$ is also, trivially, zero. Providing that $f$ is strictly positive the algorithms described above can, with appropriate changes, be applied to find the minimum of $f$. A *min-flow* is required. This may be defined:

**Definition 28** *Let $C_1$ and $C_2$ be adjacent cliques in a junction tree $\mathcal{T}$ which are joined by a separator $S_0$. Then a* **min-flow** *passing from $C_1$ (the* **source***) to $C_2$ (the* **sink***) replaces an original charge $\mathcal{K} = (\{a_C : C \in \mathcal{C}\}, \{b_S : S \in \mathcal{S}\})$, by a new charge $\mathcal{K}^* = (\{a^*_C : C \in \mathcal{C}\}, \{b^*_S : S \in \mathcal{S}\})$ , where:*

$$\begin{aligned} b^*_{S_0} &= min_{C_1 \backslash S_0} a_{C_1} \\ a^*_{C_2} &= a_{C_2} \times \lambda_{S_0} \\ \lambda_{S_0} &= \begin{cases} b^*_{S_0}(x_{S_0})/b_{S_0}(x_{S_0}) & if\, b_{S_0}(x_{S_0}) > 0 \\ 0 & if\, b_{S_0}(x_{S_0}) = 0 \end{cases} \end{aligned} \tag{3.3}$$

*and all other potentials are unaltered.*

A maximisation routine may be added to our library of *Mathematica* functions. A function which will maximise a potential table over its outermost variable is provided by `maxf`.

```
maxf[pottable_] := Block[{vars, data, newvars,
      newdata, newpot},
      vars = pottable[[1]];
      data = pottable[[2]];
      newvars = Drop[pottable[[1]], 1];
      level = Length[newvars];
      newdata = MapThread[Max, data, level];
      newpot = Join[{newvars}, {newdata}];
      Return[newpot]
]
```

The function `maxflow` passes a max-flow from a clique $a_{C_1}$, at position `pa1` in `cliques`, to a clique $a_{C_2}$, at position `pa2` in `cliques`, via a separator $b_{S_0}$, at position `pb0` in `seps`. It is identical to the `flow` function except the marginalisation (summation) function `marg` has been replaced by the maximisation function `maxf`.

```
maxflow[pa1_, pb0_, pa2_] := Block[{b0vars,diff,
      b0star,i,cur,vars,pos,ord,l0,a2star},
      b0vars = spots[[pb0,1]];
      diff = Complement[cpots[[pa1,1]],b0vars[[1]]];
      b0star = cpots[[pa1]];
      For[i=1,i<=Length[diff],i++,
         cur = diff[[i]];
         vars = b0star[[1]];
         pos = Position[vars, cur][[1,1]];
         ord = Join[{cur}, Drop[vars, {pos}]];
         b0star = rearrange[b0star, ord];
         b0star = maxf[b0star];
      ];
      l0 = div[b0star, spots[[pb0]]];
      l0 = rearrange[l0, cpots[[pa2,1]]];
      a2star = mult[cpots[[pa2]],l0];
      spots[[pb0]] = Join[{b0vars}, {b0star[[2]]}];
      cpots[[pa2]] = a2star;
]
```

The `maxprop` function propagates a schedule of max-flows. It is identical to the `prop` function except the sum-flow function `flow` has been replaced by the max-flow function `maxflow`.

```
maxprop[sched_]:=Block[{i,a1,b0,a2,pa1,pb0,pa2},
      For[i=1,i<=Length[sched],i++,
        a1 = sched[[i,1]];
        b0 = sched[[i,2]];
        a2 = sched[[i,3]];
        pa1 = Position[cliques,a1][[1,1]];
        pb0 = Position[seps,b0][[1,1]];
        pa2 = Position[cliques,a2][[1,1]];
        maxflow[pa1, pb0, pa2];
      ];
      For[i=Length[sched],i>=1,i--,
        a1 = sched[[i,3]];
        b0 = sched[[i,2]];
        a2 = sched[[i,1]];
        pa1 = Position[cliques,a1][[1,1]];
        pb0 = Position[seps,b0][[1,1]];
        pa2 = Position[cliques,a2][[1,1]];
        maxflow[pa1, pb0, pa2];
      ];
]
```

The normalisation function `maxnorm` determines $\hat{f}$, the maximum value of $f$, this it stores as the global variable `mconst`. The function then divides every clique and separator by `mconst`. The position of ones in the potential tables will then indicate the most probable configuration of all the variables.

```
maxnorm := Block[{pot,vars,i,data},
      pot = cpots[[1]];
      vars = pot[[1]];
      For[i=1,i<=Length[vars],i++,
         pot = maxf[pot];
      ];
      mconst = pot[[2]];
      For[i=1,i<=Length[cliques],i++,
         vars = cpots[[i,1]];
         data = cpots[[i,2]]/mconst;
         cpots[[i]] = Join[{vars},{data}];
      ];
      For[i=1,i<=Length[seps],i++,
         vars = spots[[i,1]];
         data = spots[[i,2]]/mconst;
         spots[[i]] = Join[{vars},{data}];
      ];
   ]
```

Under no evidence applying the `initialise`, `maxprop` and `maxnorm` functions the most probable configuration of evidence may be found to be $\bar{a}\bar{b}\bar{d}\bar{e}\bar{l}\bar{s}\bar{t}\bar{x}$ with probability 0.290362. Suppose a patient is known to have been to Asia, have dyspnoea and a positive X-ray then the most likely explanation of this evidence may be determined by application of the `initialise`, `enterev` (three times), `maxprop`, and `maxnorm` functions. The clique potentials are then found to be:

```
{ {{alpha,tau},{{0,0},{1,0.473684}}},
  {{epsilon,xi},{{0,0.408163},{0,1}}},
  {{epsilon,lambda,tau},{{{0.408163,0},{0,0}},
                         {{0,0.473684},{1,0.0526316}}}},
  {{beta,epsilon,lambda},{{{0.0654762,0},{0.472807,0.518519}},
                          {{0.408163,0},{0.473684,1}}}},
{{beta,lambda,sigma},
                 {{{0.472807,0.245614},{0.0907407,0.518519}},
                  {{0.260526,0.473684},{0.05,1}}}},
{{beta,delta,epsilon},{{{0,0},{0.0654762,0.518519}},
                       {{0,0},{0.408163,1}}}} }
```

Similarly the separator potentials are:

```
{ {{{tau},{1}},{1,0.473684}},
  {{{epsilon},{2}},{0.408163,1}},
  {{{epsilon,lambda},{3}},{{0.408163,0},{0.473684,1}}},
  {{{beta,lambda},{4}},{{0.472807,0.518519},{0.473684,1}}},
  {{{beta,epsilon},{5}},{{0.0654762,0.518519},{0.408163,1}}} }
```

The location of ones indicates that the most probable configuration is $abdelst\bar{x}$. The maximum value in the system following evidence entry and the propagation of a schedule of max-flows is found to be 0.00025137. Passing a schedule of sum-flows and normalising (by use of **prop** and **norm**) we may determine the normalisation constant, and hence the probability of the evidence, to be 0.000988. $\hat{f}$, the maximum level of $f$, under the evidence is therefore 0.25436 ( = 0.00025137/0.000988 ). Alternatively we could have passed the sum-flows following evidence entry and prior to the passing of the max-flows in order to obtain the same information more directly.

### 3.7.2 Simulation

We shall now consider a way in which, given a set of evidence $\mathcal{E}$, random realisations of the unobserved variables in the system may be simulated according to their joint conditional distribution given $\mathcal{E}$. We will let $\xi_A$ denote a set of simulated values of $X_A$ for $A \subseteq K$.

First assume that the system has been initialised and any evidence vectors necessary to enter the evidence $\mathcal{E}$ have been applied to the system. The representation $\mathcal{K}^*$ on the system now holds for a function $f^*$, where $f^*(x) = P(X = x\&\mathcal{E})$. We now apply the collect evidence phase of a propagation schedule to the system using sum-flows. Suppose the last flow in the schedule passed into a clique $C_0$ then the potential, $a_{C_0}$, on $C_0$ is proportional to the distribution of the variables $X_{C_0}$ contained in $C_0$ given $\mathcal{E}$. We may use this distribution to simulate a set of values $\xi_{C_0}$ for the variables $X_{C_0}$ given $\mathcal{E}$. We may now pass a modified distribution phase of the propagation schedule.

Consider a flow from a clique $C_1$ to a clique $C_2$ via a separator $S_0$. For the first flow of the distribution phase $C_1 = C_0$. We will, for each flow, already have simulated a set of values $\xi_{C_1}$ for the variables $X_{C_1}$ in $C_1$ given $\mathcal{E}$. Since the variables

in $S_0$ form a subset of the variables in $C_1$ we may extract $\xi_{S_0}$ from $\xi_{C_1}$ directly. Now pass the flow from $C_1$ to $C_2$ via $S_0$. The potential $a_{C_2}$ on $C_2$ is proportional to the distribution of the variables $X_{C_2}$ contained in $C_2$ given $\mathcal{E}$. Let $A = C_2 \backslash S_0$ then we may define the restricted potential $a_A$ on $A$ by $a_A(x_A) = a_{C_2}(y)$ where $y \in \chi_{C_2}$ is such that $y_A = x_A$ and $y_{S_0} = \xi_{S_0}$. In otherwords we apply the current set of simulated values, $\xi_{S_0}$, associated with $C_2$ to the potential table $a_{C_2}$ for $C_2$ and obtain a restricted potential table $a_A$ which applies to the set of variables $X_A$ we have not simulated yet. In practice $a_A$ may be formed from $a_{C_2}$ by dropping those layers of the potential table for $a_{C_2}$ which do not satisfy $\xi_{S_0}$. We may then simulate the values $\xi_A$ using $a_A$ and form $\xi_{C_2} = \{\xi_A, \xi_{S_0}\}$. Proceeding in this way we pass a full propagation schedule and generate a value for every variable from the desired joint distribution.

```
simprop[sched_]:=Block[{i,a1,b0,a2,pa1,pb0,pa2},
For[i=1,i<=Length[sched],i++,
   a1 = sched[[i,1]];
   b0 = sched[[i,2]];
   a2 = sched[[i,3]];
   pa1 = Position[cliques,a1][[1,1]];
   pb0 = Position[seps,b0][[1,1]];
   pa2 = Position[cliques,a2][[1,1]];
   flow[pa1, pb0, pa2]];
simclq = Table[{},{Length[cliques]}];
simsep = Table[{},{Length[seps]}];
simclq[[pa2]] = sim[cpots[[pa2]]];
For[i=Length[sched],i>=1,i--,
   a1 = sched[[i,3]];
   b0 = sched[[i,2]];
   a2 = sched[[i,1]];
   pa1 = Position[cliques,a1][[1,1]];
   pb0 = Position[seps,b0][[1,1]];
   pa2 = Position[cliques,a2][[1,1]];
   flow[pa1, pb0, pa2];
   simsep[[pb0]] = simsepf[pa1,pb0];
   simclq[[pa2]] = simclqf[pb0,pa2]];
]
```

We describe a set of *Mathematica* functions which will perform this simulation strategy. We assume that `initialise` and `enterev` have first been applied. The function `simprop` passes a collect evidence schedule of flows defined by `sched`. It uses the function `flow`, defined above, to facilitate this. `simclq` and `simsep` are defined to be the lists which will hold the simulated values for each clique and separator respectively. They are ordered the same as `cliques` and `seps`. The function `sim` is used to simulate the variables in the last sink clique of `sched`. Then for each stage in the distribute evidence phase a flow is passed using `flow`, the simulated values in the separator are found using `simsepf` and the simulated values for the sink clique are generated using `simclqf`.

The function `simsepf` is called with the position, `pa1`, in `cliques` of the variables in the source clique and the position, `pb0`, in `seps` of the separator. `a1list` is the list of variable and simulated value pairs for the variables in the source clique. The function extracts those pairs whose variables exist in the separator and returns a list of the corresponding pairs of variables and simulated values.

```
simsepf[pa1_,pb0_]:= Block[{a1list,a1vars,b0vars,
        list,i,pos},
        a1list = simclq[[pa1]];
        a1vars = Transpose[a1list][[1]];
        b0vars = seps[[pb0,1]];
        list = {};
        For[i=1,i<=Length[b0vars],++i,
          pos = Position[a1vars,b0vars[[i]]][[1,1]];
          list = Append[list,a1list[[pos]]]
        ];
        Return[list];
]
```

The function `simclqf` is called with the position, `pb0`, of the separator in `seps` and the position, `pa2`, of the sink clique in `cliques`. `pot` is the potential of the sink clique, and `sepev` is the list of variable and simulated value pairs defined for the separator. For each variable, `var`, in the separator the potential `pot` is rearranged such that the outermost layer of the potential table in `pot` corresponds to the variable `var`. The simulated value on that variable is then used to select that part of the potential table for which the simulated value holds. `pot` is then a potential function on one less variable - i.e. it is no longer a function of `var`.

Looping in this way over all the variables in sepev we force pot to be a potential table for the variables in the sink clique which have not yet been simulated. The function sim is used to simulate these values. Finally the list of variable and simulated value pairs for all the variables in the sink clique is returned by simclqf.

```
simclqf[pb0_,pa2_]:= Block[{pot,sepev,pvars,i,
var,lev,ord,pdata,list},
pot = cpots[[pa2]];
sepev = simsep[[pb0]];
pvars = pot[[1]];
For[i=1,i<=Length[sepev],++i,
   var = sepev[[i,1]];
   lev = sepev[[i,2]]+1;
   pvars = Complement[pvars,{var}];
   ord = Flatten[{var, pvars}];
   pot = rearrange[pot, ord];
   pdata = pot[[2,lev]];
   pot = {pvars, pdata}
];
list = Join[sepev, sim[pot]];
Return[list];
]
```

The function sim is used to simulate a set of values for a potential pot. The potential pot consists of two parts, the first is an ordered list of the variables in pot. The second part is a potential table arranged in the same order as the variable list, in which the outermost layer corresponds to the first variable in the list. The potential table need not be normalised. The potential table data is flattened and a cumulative list, pt, of the potentials is formed. The last value in this list is the reciprocal of the normalisation constant for the potential table. A Uniform(0, 1) random variable is simulated using the Random function. This value is then multiplied by the reciprocal of the normalisation constant to form the variable rand. The cumulative list of potentials allows us to determine a range of values for each combination of the variables in pot such that if rand falls within this range then the corresponding combination of the variables defines the list of simulated variables. pos determines the position of the range containing rand. The remaining code determines the list of variable and simulated value pairs corresponding to the range. This list is returned by the function.

```
sim[pot_] := Block[{vars,data,pt,ptlow,pthi,rand,
      truel,trueh,posl,posh,pos,locnums,i,lpos,
      varlist,tot,loclev},
      vars = pot[[1]];
      data = pot[[2]];
      pt = FoldList[Plus,0,Flatten[data]];
      ptlow = Drop[pt, -1];
      pthi = Drop[pt, 1];
      rand = Last[pt]*Random[];
      truel = Map[Negative, ptlow-rand];
      trueh = Map[NonNegative, pthi-rand];
      posl = Flatten[Position[truel, True]];
      posh = Flatten[Position[trueh, True]];
      pos = Intersection[posl,posh][[1]];
      locnums = {};
      For[i=1,i<=Length[vars],++i,
        lpos = Position[globvars,vars[[i]]][[1]];
        locnums = Append[locnums, globnums[[lpos]]]
      ];
      locnums = Flatten[locnums];
      varlist = {};
      tot = Apply[Times,locnums];
      For[i=1,i<=Length[vars],++i,
        tot=tot/locnums[[i]];
        loclev=Quotient[pos,tot];
        pos=Mod[pos,tot];
        If [(pos==0),
          loclev=loclev-1;
          pos = tot,
          loclev=loclev];
        varlist = Append[varlist,loclev]
      ];
      varlist = Transpose[{vars,varlist}];
      Return[varlist];
]
```

Table 3.2 gives the means and variances of 1000 simulations of the variables in the chest clinic example determined using this code. The expected mean and variance is presented in brackets. Since the variables are Bernoulli the mean corresponds to the marginal probability that a variable is true, and the variance to the product of the probability that the variable is true and the probability that the variable is false.

| Variable | Mean | Variance | Variable | Mean | Variance |
|---|---|---|---|---|---|
| $\alpha$ | 0.007<br>(0.010) | 0.006951<br>(0.009900) | $\lambda$ | 0.058<br>(0.055) | 0.054636<br>(0.051975) |
| $\beta$ | 0.458<br>(0.450) | 0.248236<br>(0.247500) | $\sigma$ | 0.500<br>(0.500) | 0.250000<br>(0.250000) |
| $\delta$ | 0.436<br>(0.436) | 0.245904<br>(0.245900) | $\tau$ | 0.012<br>(0.010) | 0.011856<br>(0.010292) |
| $\epsilon$ | 0.068<br>(0.065) | 0.063376<br>(0.060625) | $\xi$ | 0.121<br>(0.110) | 0.106359<br>(0.098126) |

Table 3.2: Means and Variances of 1000 simulations of the variables in the chest clinic example, with expected values in brackets.

## 3.8 Mixed Graphical Association Models

We now turn our attention to an exact numeric technique which builds on the discrete exact case and will enable the incorporation of continuous variables into the system. Lauritzen & Wermuth (1989) define and investigate a class of statistical models which is suitable for this task. These they term the *Conditional Gaussian distributions* (*CG-distributions*). *CG-distributions* have the property that the conditional distribution of the set of quantitative variables, given the qualitative, is multivariate Gaussian. The qualitative variables of *CG-distributions*, like those in the discrete exact case, must take a finite number of states and may thus be expressed in terms of a probability table.

Lauritzen (1992) describes a computational scheme which enables the probabilities of the qualitative variables and the means and variances of the quantitative

variables in an expert system of *CG-type* to be modelled. We present and prove Lauritzen's results and describe an implementation of his scheme using the computer language *Mathematica*. Similar implementations have also been made for *HUGIN* (Olesen, 1991) and *CAPRES* (Gammerman *et al.*, 1991). We also describe routines which facilitate simulation in this mixed case.

## 3.9 CG-Distributions and CG-Potentials

Let us first consider the distributions with which we are concerned. Suppose we have an independence graph $\mathcal{G} = (K, E)$ for a set of random variables, $X = (X_1, X_2, \ldots, X_k)$. Let us partition $K$ into a set of discrete vertices $\Delta$ and a set of continuous vertices $\Gamma$ such that $K = \Delta \cup \Gamma$. A typical element of the joint state space of discrete and continuous variables may be written in terms of its quantitative and qualitative components thus:

$$x = (x_a)_{a \in K} = (i, y) = ((i_\delta)_{\delta \in \Delta}, (y_\gamma)_{\gamma \in \Gamma})$$

where $i_\delta$ are qualitative values and $y_\gamma$ are real-valued. Each particular combination $i = (i_\delta)_{\delta \in \Delta}$ is termed a *cell* and is akin to the cells of a contingency table formed by the qualitative variables.

**Definition 29 CG-Distribution:** *We term the joint density, $f_A$, of the variables $X_A$ for $A \subseteq K$ a* **CG-distribution**, *if:*

$$f_A(x_A) = f_A(i_A, y_A) = \chi(i_A)\, exp\{g(i_A) + h(i_A)^T y_A - \frac{1}{2} y_A^T J(i_A) y_A\} \qquad (3.4)$$

*where $\chi(i_A) \in \{0, 1\}$ is an indicator function equalling one when $f_A$ is positive at $i_A$, and zero otherwise. When $\chi(i_A) \equiv 0$ we need not explicitly define $g(i_A)$ , $h(i_A)$ or $J(i_A)$.*

**Theorem 16** *Equation 3.4 is equivalent to the statement:*

$$Y_A \mid (I_A = i_A) \sim N_{|\Gamma_A|}(\xi(i_A), \Sigma(i_A)) \quad whenever \quad p(i_A) = P(I_A = i_A) > 0$$

*where we define:*

$$
\begin{aligned}
p(i_A) &= (2\Pi)^{1/2|\Gamma_A|}\{detJ(i_A)\}^{-1/2}\, exp\{g(i_A) + \frac{1}{2}h(i_A)^T J(i_A)^{-1}h(i_A)\} \\
\xi(i_A) &= J(i_A)^{-1}h(i_A) \\
\Sigma(i_A) &= J(i_A)^{-1}, \quad \textit{and} \quad \Sigma \quad \textit{is positive definite.}
\end{aligned} \tag{3.5}
$$

**Proof.** Using Equation 3.5 we may show that:

$$
\begin{aligned}
f_A(x_A) &= f_A(i_A, y_A) \\
&= \chi(i_A)\exp\left\{g(i_A) + h(i_A)^T y_A - \frac{1}{2}y_A^T J(i_A) y_A\right\} \\
&= \chi(i_A)\exp\left\{\log p(i_A) + \frac{1}{2}\log\det\Sigma(i_A)^{-1} - \frac{1}{2}\mid\Gamma_A\mid\log(2\Pi)\right. \\
&- \left.\frac{1}{2}\xi(i_A)^T\Sigma(i_A)^{-1}\xi(i_A) + \left(\Sigma(i_A)^{-1}\xi(i_A)\right)^T y_A - \frac{1}{2}y_A^T\Sigma(i_A)^{-1}y_A\right\} \\
&= \chi(i_A)p(i_A)\{\det\Sigma(i_A)\}^{-1/2}(2\Pi)^{-1/2|\Gamma_A|} \\
&\times exp\left\{-\frac{1}{2}\left(\xi(i_A)^T\Sigma(i_A)^{-1}\xi(i_A) - 2\xi(i_A)^T\Sigma(i_A)^{-1}y_A + y_A^T\Sigma(i_A)^{-1}y_A\right)\right\} \\
&= \chi(i_A)p(i_A)\{\det\Sigma(i_A)\}^{-1/2}(2\Pi)^{-1/2|\Gamma_A|} \\
&\times \exp\left\{-\frac{1}{2}(y_A - \xi(i_A))^T\Sigma(i_A)^{-1}(y_A - \xi(i_A))\right\} \\
&= P(I_A = i_A)f(Y_A \mid I_A = i)
\end{aligned}
$$

Thus we see that the conditional distribution of the set of quantitative variables, $Y_A$, given the qualitative variables, $I_A$, is multivariate Gaussian.

□

We may define two triples for $\chi(i_A) > 0$. The *canonical characteristics* $(g, h, J)$ and the *moment characteristics* $(p, \xi, \Sigma)$. Each triple may be used to fully describe the joint distribution function, $f_A$, of the set of random variables $X_A$. In the purely discrete case, where all variables are qualitative the canonical characteristics are $(g, ., .)$ since $h$ and $J$ are of zero shape and size. Similarly, where all the variables are real-valued, the canonical characteristics are $(., h, J)$. We notice that it is possible to convert from the moment characteristics to the canonical characteristics using the formulae given in Equations 3.6. Similarly the canonical characteristics may be converted to the moment characteristics using the formulae in Equations 3.5.

$$
\begin{aligned}
g(i_A) &= \log p(i_A) + \frac{1}{2}\{\log\det\Sigma(i_A)^{-1} - \mid \Gamma_A \mid \log(2\Pi) - \xi(i_A)^T\Sigma(i_A)^{-1}\xi(i_A)\} \\
h(i_A) &= \Sigma(i_A)^{-1}\xi(i_A) \\
J(i_A) &= \Sigma(i_A)^{-1}
\end{aligned}
\tag{3.6}
$$

A more general representation of a CG-distribution for a set of random variables $X_A$, $A \subseteq K$, is any function $\phi_A$ of the form:

$$
\phi_A(x_A) = \phi_A(i_A, y_A) = \chi(i_A)\, exp\{g(i_A) + h(i_A)^T y_A - \frac{1}{2} y_A^T J(i_A) y_A\} \tag{3.7}
$$

where $J(i_A)$ is symmetric and $\phi_A$ is not necessarily a density. We term $\phi_A$ a *CG-potential*. The conversion formulae of Equations 3.5 and 3.6 apply equally to CG-potentials.

In order to represent a CG-potential for a set of random variables $X_A$ in our implementation we will require two ingredients. Firstly the names of the variables the potential is representing, and secondly either the canonical or moment characteristics of the potential. As in the discrete exact case we will use *lists*, the basic data structure of *Mathematica* (Wolfram, 1991), to accomplish this. We represent a CG-potential as a list consisting of two elements. The first element is itself a list of two elements, the two elements being lists of the names of the continuous and discrete variables defined by the CG-potential.

The second element of the CG-potential data structure defines the CG-potential itself in terms of its canonical characteristics. Since the system is allowed to contain discrete variables a CG-potential will, in general, consist of a multivariate table each dimension of which corresponds to a different discrete variable. Within each dimension each level of the respective discrete variable will be represented by a separate layer. Thus the CG-potential is constructed as a nested list of lists defining a multivariate array. The outermost dimension of the array corresponds to the first discrete variable listed. The next dimension corresponds to the second discrete variable listed and so on. Thus a single cell in the array corresponds to a unique combination of the discrete variables in the potential.

Each cell is a list consisting of three elements. These three elements define the canonical characteristics of the CG-potential given the levels of the discrete

variables defining that cell. The first element is $g$, which is represented by a real number. The second element is $h$, which is a list of real numbers representing a vector whose length, $\mid \Gamma_A \mid$, is equal to the number of continuous variables in the CG-potential. The ordering of these numbers is the same as the ordering of the names of the continuous variables. The third element is $J$ , which is a list of length $\mid \Gamma_A \mid$ of lists of length $\mid \Gamma_A \mid$ of real numbers which represent the required matrix. Again, the ordering of the elements of $J$ is consistent with the ordering of continuous variable names.

To represent the indicator function $\chi$ we only keep a cell if $\chi = 1$. This is advantageous in terms of both space and processing time. Table 3.3 gives an example of a CG-potential and Figure 3.6 shows how it may be defined for our implementation.

| | $B = b$ | | | | | $B = \bar{b}$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $g$ | $h$ | | $J$ | | $g$ | $h$ | | $J$ | |
| | | | | $d$ | $e$ | | | | $d$ | $e$ |
| $W = w$ | $-700$ | $d$ | 200 | 30 | $-30$ | $-280$ | $d$ | 75 | 10 | $-10$ |
| | | $e$ | $-200$ | $-30$ | 30 | | $e$ | $-75$ | $-10$ | 10 |
| | $g$ | $h$ | | $J$ | | $g$ | $h$ | | $J$ | |
| | | | | $d$ | $e$ | | | | $d$ | $e$ |
| $W = \bar{w}$ | $-450$ | $d$ | 150 | 25 | $-25$ | $-245$ | $d$ | 70 | 15 | $-15$ |
| | | $e$ | $-150$ | $-25$ | 25 | | $e$ | $-70$ | $-15$ | 15 |

Table 3.3: The canonical characteristics of a CG-potential defined on two discrete binary variables, $B$ and $W$, and two continuous variables, $D$ and $E$.

```
{{{d, e}, {b, w}},
{{{{-700}, {200, -200}, {{30, -30}, {-30, 30}}},
{{-450}, {150, -150}, {{25, -25}, {-25, 25}}}},
{{{-280}, {75, -75}, {{10, -10}, {-10, 10}}},
{{-245}, {70, -70}, {{15, -15}, {-15, 15}}}}}}
```

Figure 3.6: The *Mathematica* representation of the CG-potential described in Table 3.3.

## 3.10 The Waste Incinerator Example

Let us consider the problem described in Lauritzen (1992) which seeks to model the emission of heavy metals from a waste incinerator at a particular point in time. We will use this problem as a standard example of the mixed case and will apply the methodology of this and later chapters to it.

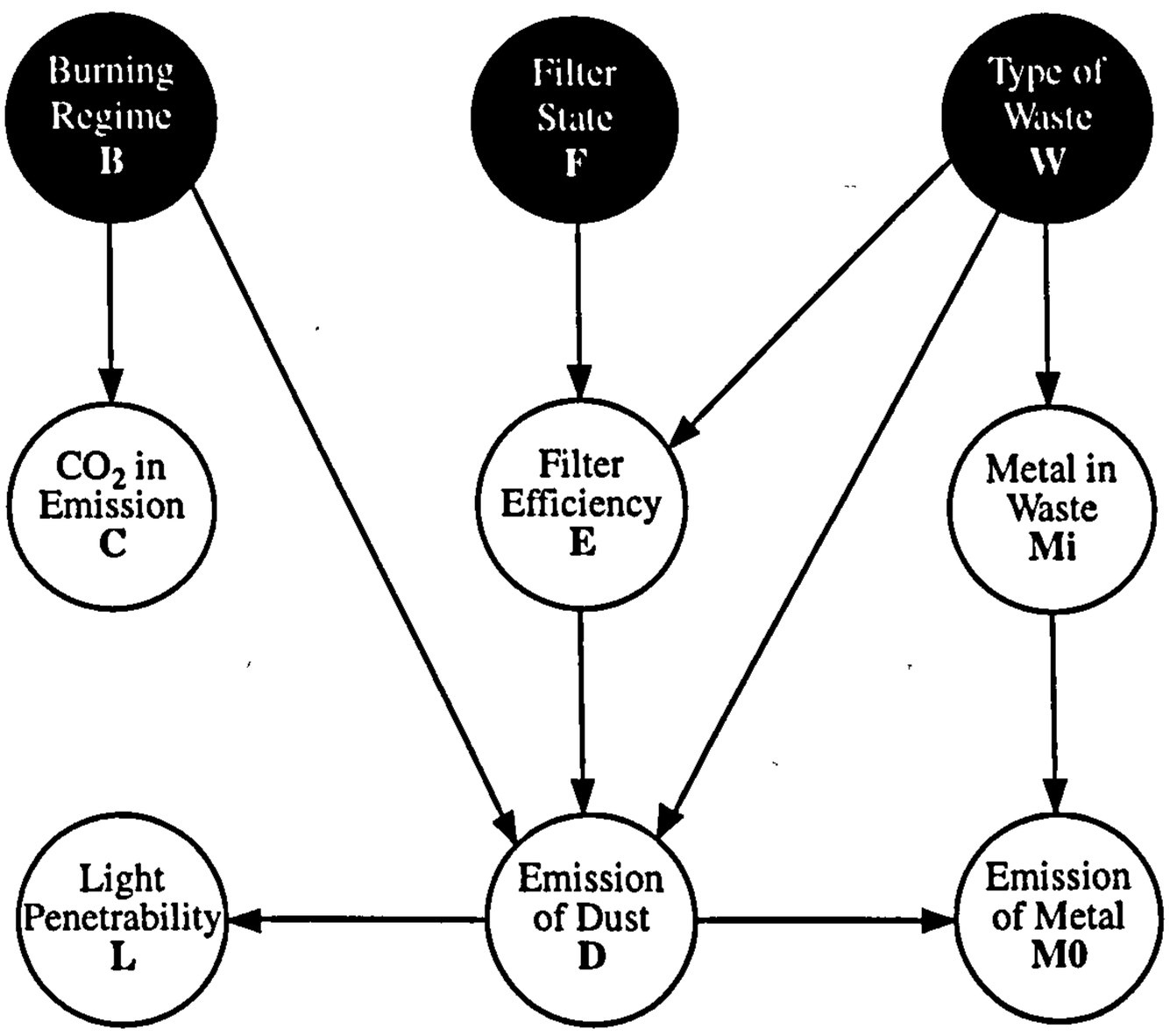



Figure 3.7: The causal probabilistic network of the waste incinerator problem.

The waste incinerator burns one of two types of waste: household or industrial. It possesses an electro-filter which filters out dust. The inefficiency of the filter is determined by its technical state, either intact or defective, and by the type of waste being incinerated. The burning regime, which may be either stable or unstable, is monitored by measuring the concentration of $CO_2$ in the emission. The burning regime controls the emission of dust as does the efficiency of the filter and the type of waste being incinerated. The emission of dust is monitored by measuring the penetrability of light. The amount of metal in the waste is determined by the type of waste. The emission of heavy metals from the incinerator depends upon the amount of metal in the waste and the amount of dust that is emitted. The CPN in Figure 3.7 represents the waste incinerator problem. The three discrete variables, burning regime ($B$), filter state ($F$) and type of waste ($W$), are shown as dots and the six continuous variables as circles.

A junction tree may be formed from the CPN. Firstly its moral graph is formed by adding undirected edges between parents that are not already linked and dropping the directions on all the other edges to produce an undirected graph. Then in a process known as *strong triangulation* further links may possibly be needed to form a *strongly decomposable* marked graph. Strong triangulation may be accomplished by triangulating using an elimination sequence of nodes that eliminate continuous nodes first. Details of such triangulation processes may be found in Kjærulff (1990). The junction tree may then be formed from the undirected graph we have generated in the usual way. The junction tree for the waste incinerator example is given in Figure 3.8.

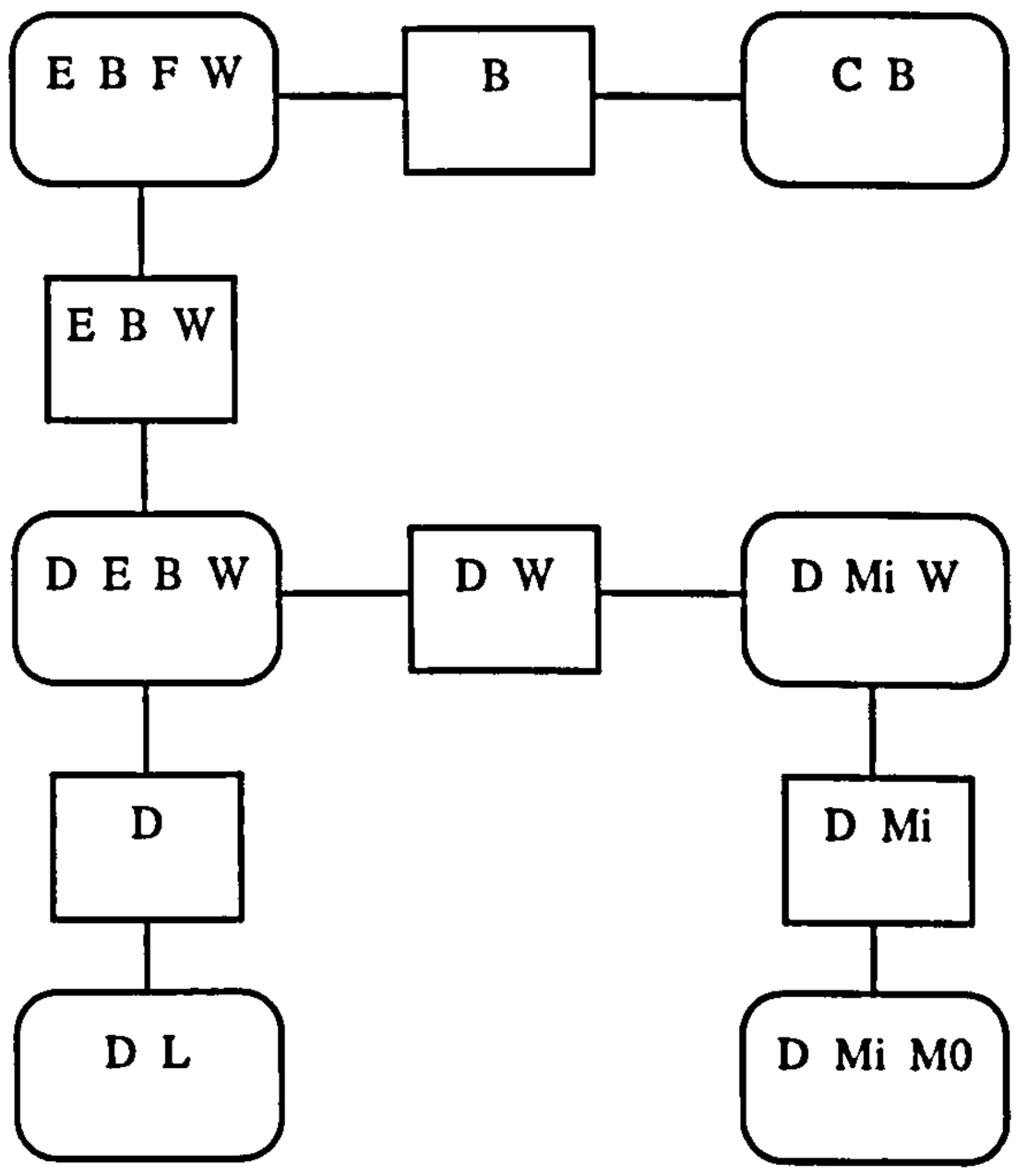



Figure 3.8: The junction tree of the waste incinerator problem. Cliques are represented by rounded boxes and separators by rectangles.

An important property of junction trees formed from decomposable marked graphs is that they possess at least one *strong root* (see Theorem 13). In the waste incinerator example there are two possible strong roots $\{C, B\}$ and $\{E, B, F, W\}$. The presence of a strong root will be shown to be important in the determination of the propagation schedule to be used.

In our implementation we must specify the cliques and the separators. Each clique is specified as a list of two lists, the first being a list of the continuous variables in the clique and the second a list of the discrete variables in the clique. We then have a list `cliques` which is a list of all the cliques. We do the same for the

separators but this time for each separator we have a list of three elements. The first two elements are identical in structure to `cliques` and define the variable names but the additional, third, element is a number. This allows us to have multiple separators with the same variables yet we are still able to determine which separator is which. We call the list of separators `seps`. The structure of the junction tree does not need to be specified directly but rather through the propagation schedule. The lists of cliques and separators are as follows:

```
cliques = {{{c}, {b}}, {{e}, {b, f, w}},
{{d, e}, {b, w}}, {{d, mi}, {w}},
{{d, l}, { }}, {{m0, d, mi} ,{ }}}

seps = {{{ }, {b}, 1}, {{e}, {b, w}, 2 },
{{d}, {w}, 3 }, {{d}, { }, 4 },
{ {d, mi}, { }, 5 }}
```

## 3.11 Defining the Variables

If we assume that no continuous nodes may have discrete children then we may consider the probability distribution of a discrete variable $I_a$ given the states of its parents, which are then ensured to be all discrete. This will produce a table of probabilities $p(i_a; i_{pa(a)})$ with each probability corresponding to a cell in the table.

In the case of a continuous node $Y_a$ with discrete parents $I_{pa(a)}$ and continuous parents $Y_{pa(a)}$, we assume the conditional distribution of $Y_a$ given its parents to be of the following type:

$$Y_a \mid X_{pa(a)} \sim N(\alpha(i_{pa(a)}) + \beta(i_{pa(a)})^T y_{pa(a)}, \gamma(i_{pa(a)})) \tag{3.8}$$

where $X_{pa(a)}$ has state space $\mathcal{I}_{pa(a)} \times \mathbb{R}^{|\Gamma_{pa(a)}|}$ and $\alpha(i_{pa(a)}) \in \mathbb{R}$, $\beta(i_{pa(a)}) \in \mathbb{R}^{|\Gamma_{pa(a)}|}$ and $\gamma(i_{pa(a)}) > 0$. Thus the mean of the conditional distribution of $Y_a$ given its parents is a linear function of the states at the continuous parent nodes, $Y_{pa(a)}$, while the variance does not depend on the continuous parent nodes at all. Both the mean and the variance may depend on the discrete parents.

**Proposition 7** *Let $Y_a \mid X_{pa(a)}$ take the form given in Equation 3.8 then the canonical characteristics of the CG-potentials defined above may be determined by using the following formulae:*

$$\begin{aligned} g^*(i_{pa(a)}) &= -\frac{\alpha(i_{pa(a)})^2}{2\gamma(i_{pa(a)})} - \frac{1}{2}\left\{log\,(2\Pi\gamma(i_{pa(a)}))\right\} \\ h^*(i_{pa(a)}) &= \frac{\alpha(i_{pa(a)})}{\gamma(i_{pa(a)})}\begin{pmatrix} 1 \\ -\beta(i_{pa(a)}) \end{pmatrix} \\ J^*(i_{pa(a)}) &= \frac{1}{\gamma(i_{pa(a)})}\begin{pmatrix} 1 & -\beta(i_{pa(a)})^T \\ -\beta(i_{pa(a)}) & \beta(i_{pa(a)})\beta(i_{pa(a)})^T \end{pmatrix} \end{aligned} \tag{3.9}$$

**Proof.** By Equation 3.8:

$$f(y_a \mid x_{pa(a)}) = \frac{1}{\sqrt{2\Pi\gamma(i_{pa(a)})}}\exp\left\{-\frac{1}{2\gamma(i_{pa(a)})}\left(y_a - \alpha(i_{pa(a)}) - \beta(i_{pa(a)})^T y_{pa(a)}\right)^2\right\}$$

$$\begin{aligned} &= \exp\left\{-\frac{1}{2}\log\left(2\Pi\gamma(i_{pa(a)})\right) - \frac{1}{2\gamma(i_{pa(a)})}\left(y_a - \alpha(i_{pa(a)}) - \beta(i_{pa(a)})^T y_{pa(a)}\right)^2\right\} \\ &= \exp\left\{-\frac{1}{2}\log(2\Pi\gamma(i_{pa(a)})) - \frac{\alpha(i_{pa(a)})^2}{2\gamma(i_{pa(a)})}\right. \\ &\quad + \frac{1}{2\gamma(i_{pa(a)})}\left[2\alpha(i_{pa(a)})y_a - \alpha(i_{pa(a)})y_{pa(a)}^T\beta(i_{pa(a)}) - \alpha(i_{pa(a)})\beta(i_{pa(a)})^T y_{pa(a)}\right] \\ &\quad - \frac{1}{2\gamma(i_{pa(a)})}\left[y_a^2 - \beta(i_{pa(a)})^T y_{pa(a)}y_a - y_{pa(a)}^T\beta(i_{pa(a)})y_a\right. \\ &\quad \left.\left. + y_{pa(a)}^T\beta(i_{pa(a)})\beta(i_{pa(a)})^T y_{pa(a)}\right]\right\} \\ &= \exp\left\{-\frac{\alpha(i_{pa(a)})^2}{2\gamma(i_{pa(a)})} - \frac{1}{2}\left[\log(2\Pi\gamma(i_{pa(a)}))\right] + \frac{\alpha(i_{pa(a)})}{\gamma(i_{pa(a)})}\left[y_a - \beta(i_{pa(a)})^T y_{pa(a)}\right]\right. \\ &\quad - \frac{1}{2\gamma(i_{pa(a)})}\left[y_a^2 - \beta(i_{pa(a)})^T y_{pa(a)}y_a - y_{pa(a)}^T\beta(i_{pa(a)})y_a\right. \\ &\quad \left.\left. + y_{pa(a)}^T\beta(i_{pa(a)})\beta(i_{pa(a)})^T y_{pa(a)}\right]\right\} \\ &= \exp\left\{-\frac{\alpha(i_{pa(a)})^2}{2\gamma(i_{pa(a)})} - \frac{1}{2}\left[\log(2\Pi\gamma(i_{pa(a)}))\right] + \frac{\alpha(i_{pa(a)})}{\gamma(i_{pa(a)})}\left(1 - \beta(i_{pa(a)})^T\right)\begin{pmatrix} y_a \\ y_{pa(a)} \end{pmatrix}\right. \\ &\quad \left. - \frac{1}{2}\begin{pmatrix} y_a & y_{pa(a)}^T \end{pmatrix}\frac{1}{\gamma(i_{pa(a)})}\begin{pmatrix} 1 & -\beta(i_{pa(a)})^T \\ -\beta(i_{pa(a)}) & \beta(i_{pa(a)})\beta(i_{pa(a)})^T \end{pmatrix}\begin{pmatrix} y_a \\ y_{pa(a)} \end{pmatrix}\right\} \\ &= \exp\left\{g^*(i_{pa(a)}) + h^*(i_{pa(a)})^T y_{a,pa(a)} - \frac{1}{2}y_{a,pa(a)}^T K^*(i_{pa(a)})y_{a,pa(a)}\right\} \end{aligned}$$

Thus the canonical characteristics, $(g^*(i_{pa(a)}), h^*(i_{pa(a)}), J^*(i_{pa(a)}))$, are as defined in Equations 3.9 above. □

| C \| B (C: C02 in Emission) | | | Mi \| W (Mi: Metal in Waste) | | |
|---|---|---|---|---|---|
| **B: Burning Regime** | $\alpha$ | $\gamma$ | **W: Type of Waste** | $\alpha$ | $\gamma$ |
| Stable | -2 | 0.1 | Industrial | 1/2 | 0.01 |
| Unstable | -1 | 0.3 | Household | -1/2 | 0.005 |

| E \| (F, W) (E: Filter Efficiency) | | | |
|---|---|---|---|
| **F: Filter State** | **W: Type of Waste** | $\alpha$ | $\gamma$ |
| Intact | Industrial | -3.9 | 0.00002 |
| | Household | -3.2 | 0.00002 |
| Defective | Industrial | -0.4 | 0.0001 |
| | Household | -0.5 | 0.0001 |

| D \| (B, E, W) (D: Emission of Dust, E: Filter Efficiency) | | | | |
|---|---|---|---|---|
| **W: Type of Waste** | **W: Type of Waste** | $\alpha$ | $\beta$ | $\gamma$ |
| Stable | Industrial | 6.5 | (1) | 0.03 |
| | Household | 6.0 | (1) | 0.04 |
| Unstable | Industrial | 7.5 | (1) | 0.10 |
| | Household | 7.0 | (1) | 0.10 |

| L \| D (L: Light Penetrability, D: Emission of Dust) | | | M0 \| D, Mi (M0: Emission of Metal, D: Emission of Dust, Mi: Metal in Waste | | |
|---|---|---|---|---|---|
| $\alpha$ | $\beta$ | $\gamma$ | $\alpha$ | $\beta$ | $\gamma$ |
| 3 | (-0.5) | 0.25 | 0 | (1, 1) | 0.002 |

Table 3.4: The conditional distributions of the continuous variables in the waste incinerator problem.

| B: Burning Regime | | F: Filter State | | W: Type of Waste | |
|---|---|---|---|---|---|
| $b$ | $P(B = b)$ | $f$ | $P(F = f)$ | $w$ | $P(W = w)$ |
| Stable | 17/20 | Intact | 19/20 | Industrial | 2/7 |
| Unstable | 3/20 | Defective | 1/20 | Household | 5/7 |

Table 3.5: The discrete distributions in the waste incinerator problem.

Proposition 7 gives us a strategy for defining each variable by specifying their conditional distributions through the values of $\alpha$, $\beta$ and $\gamma$, where appropriate, and then generating the required canonical characteristics. For the waste example we specify the conditional distributions as in Tables 3.4 and 3.5 above.

In our implementation we structure the conditional distributions in a similar way to the way that we structure the CG-potentials. Each conditional distribution consists of a list of variables and a data list. The data list consists of cells ordered by the list of variables. The cell elements are probabilities in the discrete case and $\alpha$, $\gamma$ (and $\beta$ ) in the continuous case. The conditional distributions for $B$ and $D \mid E, B, W$ are as follows:

```
condb  =  {{{},{b}},{{{.85}},{{.15}}}}

conddebw  =  {{{d,e},{b,w}},
              {{{{6.5},{0.03},{1}},{{6.0},{0.04},{1}}},
              {{{7.5},{0.10},{1}},{{7.0},{0.10},{1}}}}}
```

In addition to each conditional distribution we have a list, `condpots`, of the names of the conditional distributions and a corresponding list, `dists`, of which variable is defined by each conditional distribution. The positioning of `condpots` and `dists` is such that the distribution of the $n$-th variable in `dists` is specified by the $n$-th conditional distribution in `condpots`. These two lists are as follows:

```
condpots  =  {condb, condf, condw, condcb, condefw, conddebw,
              condmiw, condld, condm0dmi}
   dists  =  {{b},{f},{w},{c},{e},{d},{mi},{l},{m0}}
```

Using these three data structures, conditional distributions, `condpots` and `dists`, we are able to form the initial canonical characteristics as given by Equations 3.9.

## 3.12 Basic Operations

We will need to define a set of basic manipulations which we may apply to the CG-potentials. As in the discrete exact case we will require extension, multiplication, division, and marginalisation functions. The basic goal of these manipulations will be to determine a set of operations which we may use to define a propagation algorithm for this mixed case. Wherever possible we will attempt to define our manipulative operations in terms of only one set of characteristics. This will reduce the requirement to convert from one set of characteristics to the other - a process which may be both computationally expensive and result in loss of accuracy. The canonical characteristics $(g, h, J)$ turn out to be easier to manipulate than the moment characteristics $(p, \xi, \Sigma)$, however, we will be obliged to use the latter to facilitate marginalisation over any discrete variables.

It is clear, in general, that we will be unable to retain the structure of a CG-potential when marginalising over a discrete variable since we cannot determine a triple $(g', h', J')$ for which:

$$\begin{aligned} exp\{g'(i) + h'(i)^T y_A - \frac{1}{2} y_A^T J'(i) y_A\} &= exp\{g_1(i) + h_1(i)^T y_A - \frac{1}{2} y_A^T J_1(i) y_A\} \\ &+ exp\{g_2(i) + h_2(i)^T y_A - \frac{1}{2} y_A^T J_2(i) y_A\} \end{aligned}$$

The result is instead a mixture of CG-potentials. We can, however, ensure that the moment characteristics of any derived function are preserved by any manipulative operation and determine unique canonical characteristics from them by Equation 3.6. We will now define the required set of manipulative functions.

### 3.12.1 Extending a CG-Potential

**Definition 30 Extension of a CG-Potential:** *Let $U \subseteq V \subseteq K$ and $\phi(x_U) = \phi(i_U, y_U)$ be a CG-potential defined on $\mathcal{X}_U = \mathcal{I}_U \times \mathcal{Y}_U$ with canonical characteristics $(g, h, J)$. We define $\eta(x_V) = \eta(i_V, y_V)$ to be the* **extension** *of $\phi(x_U)$ to $V$, where $\eta(x_V)$ is a CG-potential defined on $\mathcal{X}_V = \mathcal{I}_V \times \mathcal{Y}_V = (\mathcal{I}_U \times \mathcal{I}_{V \setminus U}) \times (\mathcal{Y}_U \times \mathcal{Y}_{V \setminus U})$ with canonical characteristics $(g', h', J')$ where:*

$$
\begin{aligned}
g'(i_U, i_{V\setminus U}) &= g(i_U) \\
h'(i_U, i_{V\setminus U}) &= \begin{pmatrix} h(i_U) \\ \{0\} \end{pmatrix} \\
J'(i_U, i_{V\setminus U}) &= \begin{pmatrix} J(i_U) & \{0\} \\ \{0\} & \{0\} \end{pmatrix} \qquad (3.10)
\end{aligned}
$$

*and each* $\{0\}$ *represents a vector or matrix of appropriate dimension for which each element is a zero.*

This simply gives us a way of extending the canonical characteristics of CG-potentials so that they occupy the same space as each other. This will aid in their combination. In our implementation we may also need to rearrange the data so that both the discrete and the continuous variables of two CG-potentials are in the same order so that we may combine them. This is a relatively simple task in *Mathematica*. We have two rearrangement functions, one which rearranges the discrete variables and corresponding data and one which rearranges the continuous variables and corresponding data. Where it is clear from the context we will not distinguish between a CG-potential and its extension.

### 3.12.2 Multiplying Two CG-Potentials

**Definition 31 Multiplication of two CG-Potentials:** *Let* $\phi$ *and* $\eta$ *be two CG-potentials defined on the spaces* $\mathcal{X}_U = \mathcal{I}_U \times \mathcal{Y}_U$ *and* $\mathcal{X}_V = \mathcal{I}_V \times \mathcal{Y}_V$ *respectively with* $U \subseteq K$ *and* $V \subseteq K$. *The* **multiplication** *of* $\phi$ *and* $\eta$, *denoted* $\phi \times \eta$, *defined on the space* $\mathcal{X}_{U\cup V} = \mathcal{I}_{U\cup V} \times \mathcal{Y}_{U\cup V}$ *is defined to be:*

$$
(\phi \times \eta)(x_{U\cup V}) = \phi(x_{U\cup V})\eta(x_{U\cup V}) \qquad (3.11)
$$

*where* $\phi$ *and* $\eta$ *on the right-hand side have first been extended to occupy* $\mathcal{X}_{U\cup V}$.

**Theorem 17** *Let* $\phi$ *and* $\eta$ *be two CG-potentials extended to occupy the same space* $\mathcal{X}_V = \mathcal{I}_V \times \mathcal{Y}_V$ *where* $V \subseteq K$. *Suppose that* $\phi$ *and* $\eta$ *have canonical characteristics* $(g_1, h_1, J_1)$ *and* $(g_2, h_2, J_2)$, *respectively, then their product* $\phi \times \eta$ *is a CG-potential with canonical characteristics* $(g', h', J')$ *where:*

$$
(g', h', J') = (g_1 + g_2, h_1 + h_2, J_1 + J_2) \qquad (3.12)
$$

**Proof.**

$$\begin{aligned}
\phi(x_V) \times \eta(x_V) &= \chi_1(iv) \exp\left\{g_1(iv) + h_1(iv)^T y_V - \frac{1}{2} y_V^T J_1(iv) y_V\right\} \\
&\times \chi_2(iv) \exp\left\{g_2(iv) + h_2(iv)^T y_V - \frac{1}{2} y_V^T J_2(iv) y_V\right\} \\
&= \Big(\chi_1(iv) \times \chi_2(iv)\Big) \exp\left\{\Big(g_1(iv) + g_2(iv)\Big)\right. \\
&+ \Big(h_1(iv) + h_2(iv)\Big)^T y_V - \left.\frac{1}{2} y_V^T \Big(J_1(iv) + J_2(iv)\Big) y_V\right\} \\
&= \chi'(iv) \exp\left\{g'(iv) + h'(iv)^T y_V - \frac{1}{2} y_V^T J'(iv) y_V\right\}
\end{aligned}$$

Thus the product forms a CG-potential with the canonical characteristics as given in Equation 3.12.

□

To form the product of two CG-potentials in *Mathematica* we simply apply Equation 3.12 to each cell of the rearranged extensions of the two CG-potentials.

### 3.12.3 Dividing Two CG-Potentials

**Definition 32 Division of two CG-Potentials:** *Let $\phi$ and $\eta$ be two CG-potentials defined on the spaces $\mathcal{X}_U = \mathcal{I}_U \times \mathcal{Y}_U$ and $\mathcal{X}_V = \mathcal{I}_V \times \mathcal{Y}_V$ respectively with $U \subseteq K$ and $V \subseteq K$. The* **division** *of $\phi$ by $\eta$, denoted $\phi/\eta$, defined on the space $\mathcal{X}_{U \cup V} = \mathcal{I}_{U \cup V} \times \mathcal{Y}_{U \cup V}$ is defined to be:*

$$(\phi/\eta)(x_{U \cup V}) = \begin{cases} 0 & if \phi(x_{U \cup V}) = 0 \\ (\phi(x_{U \cup V})/\eta(x_{U \cup V})) & if \eta(x_{U \cup V}) \neq 0 \\ undefined & otherwise \end{cases} \tag{3.13}$$

*where $\phi$ and $\eta$ on the right-hand side have first been extended to occupy $\mathcal{X}_{U \cup V}$.*

**Theorem 18** *Let $\phi$ and $\eta$ be two CG-potentials extended to occupy the same space $\mathcal{X}_V = \mathcal{I}_V \times \mathcal{Y}_V$ where $V \subseteq K$ and let $\phi(x_V) \neq 0$ and $\eta(x_V) \neq 0$. Suppose that $\phi$ and $\eta$ have canonical characteristics $(g_1, h_1, J_1)$ and $(g_2, h_2, J_2)$, respectively, then $\phi/\eta$, the division of $\phi$ by $\eta$, is a CG-potential with canonical characteristics $(g', h', J')$ where:*

$$(g', h', J') = (g_1 - g_2, h_1 - h_2, J_1 - J_2) \tag{3.14}$$

**Proof.**

$$\begin{aligned}
\phi(x_V)/\eta(x_V) &= \chi_1(i_V)\exp\left\{g_1(i_V)+h_1(i_V)^T y_V-\frac{1}{2}y_V^T J_1(i_V)y_V\right\}\\
&/\ \chi_2(i_V)\exp\left\{g_2(i_V)+h_2(i_V)^T y_V-\frac{1}{2}y_V^T J_2(i_V)y_V\right\}\\
&= \left(\frac{\chi_1(i)}{\chi_2(i)}\right)\exp\left\{\left(g_1(i_V)-g_2(i_V)\right)\right.\\
&+ \left.\left(h_1(i_V)-h_2(i_V)\right)^T y-\frac{1}{2}y_V^T\left(J_1(i_V)-J_2(i_V)\right)y_V\right\}\\
&= \chi'(i_V)\exp\left\{g'(i_V)+h'(i_V)^T y_V-\frac{1}{2}y_V^T J'(i_V)y_V\right\}
\end{aligned}$$

Thus the division forms a CG-potential with the canonical characteristics as given in Equation 3.14.

□

In order to determine the division of two CG-Potentials in *Mathematica* we apply Equation 3.14 and, where necessary, Equation 3.14 to each cell of the rearranged extensions of the two CG-potentials.

### 3.12.4 Marginalising over Discrete Variables

**Definition 33 Discrete Marginalisation of a CG-Potential:** *Let $U \subseteq V \subseteq K$ and let $\phi(x_V) = \phi(i_V, y_V) = \phi(i_U, i_{V\setminus U}, y_V)$ be a CG-potential defined on $\mathcal{X}_V = \mathcal{I}_V \times \mathcal{Y}_V = (\mathcal{I}_U \times \mathcal{I}_{V\setminus U}) \times \mathcal{Y}_V$. Then we will let the expression $\sum_{V\setminus U}\phi(x_V)$ denote the* **marginalisation** *of $\phi(x_V)$, with respect to $I_{V\setminus U}$, to a function $\eta(x_U)$ defined on the space $\mathcal{X}_U = \mathcal{I}_U \times \mathcal{Y}_V$ where:*

$$\eta(x_U) = \sum_{V\setminus U}\phi(x_V) = \sum_{i_{V\setminus U}}\phi(x_V) \tag{3.15}$$

We may distinguish two cases when we are required to marginalise over discrete variables. The first case allows marginalisation to be performed by the manipulation of the canonical characteristics alone. It results in the formation of a CG-potential. The second case requires the use of the moment characteristics and does not form a CG-potential.

**Theorem 19** *Let $U \subseteq V \subseteq K$ and $\phi(x_V) = \phi(i_V, y_V) = \phi(i_U, i_{V\setminus U}, y_V)$ be a CG-potential defined on $\mathcal{X}_V = \mathcal{I}_V \times \mathcal{Y}_V = (\mathcal{I}_U \times \mathcal{I}_{V\setminus U}) \times \mathcal{Y}_V$ with canonical characteristics $(g, h, J)$. If $h$ and $J$ do not depend on $i_{V\setminus U}$ then we may write:*

$$(g(i_V), h(i_U, i_{V\setminus U}), J(i_U, i_{V\setminus U})) = (g(i_V), h(i_U), J(i_U)) \tag{3.16}$$

*and the marginalisation of $\phi(x_V)$, with respect to $I_{V\setminus U}$, to a function $\eta(x_U)$ is a CG-potential with canonical characteristics $(g', h', J')$ where:*

$$\begin{aligned} g'(i_U) &= log\Big\{ \sum_{\substack{i_{V\setminus U}: \\ \chi(i_U, i_{V\setminus U})=1}} exp\left\{g(i_U, i_{V\setminus U})\right\}\Big\} \\ h'(i_U) &= h(i_U) \\ J'(i_U) &= J(i_U) \end{aligned} \tag{3.17}$$

**Proof.**

$$\begin{aligned} \eta(x_U) &= \sum_{i_{V\setminus U}} \phi(i_U, i_{V\setminus U}, y_V) \\ &= \sum_{i_{V\setminus U}} \chi(i_U, i_{V\setminus U}) \exp\Big\{g(i_U, i_{V\setminus U}) + h(i_U, i_{V\setminus U})^T y - \frac{1}{2} y^T J(i_U, i_{V\setminus U}) y\Big\} \\ &= \sum_{i_{V\setminus U}} \chi(i_U, i_{V\setminus U}) \exp\Big\{g(i_U, i_{V\setminus U}) + h(i_U)^T y - \frac{1}{2} y^T J(i_U) y\Big\} \\ &= \Big(\sum_{i_{V\setminus U}} \chi(i_U, i_{V\setminus U}) \exp\Big\{g(i_U, i_{V\setminus U})\Big\}\Big)\Big(\exp\Big\{h(i_U)^T y - \frac{1}{2} y^T J(i_U) y\Big\}\Big) \\ &= \chi'(i_U)\Big(\sum_{\substack{i_{V\setminus U}: \\ \chi(i_U, i_{V\setminus U})=1}} \exp\Big\{g(i_U, i_{V\setminus U})\Big\}\Big)\Big(\exp\Big\{h(i_U)^T y - \frac{1}{2} y^T J(i_U) y\Big\}\Big) \\ &= \chi'(i_U)\exp\Big\{\log\Big(\sum_{\substack{i_{V\setminus U}: \\ \chi(i_U, i_{V\setminus U})=1}} \exp\Big\{g(i_U, i_{V\setminus U})\Big\}\Big) + h(i_U)^T y - \frac{1}{2} y^T J(i_U) y\Big\} \\ &= \chi'(i_U)\exp\Big\{g'(i_U) + h'(i_U)^T y - \frac{1}{2} y^T J'(i_U) y\Big\} \end{aligned}$$

where:

$$\chi'(i_U) = \begin{cases} 0 & \text{if } \chi(i_U, i_{V\setminus U}) = 0 \text{ for all } i_{V\setminus U} \\ 1 & \text{otherwise} \end{cases} \tag{3.18}$$

Thus $\eta(x_U)$ is a CG-potential with canonical characteristics as given in Equation 3.17.

□

**Theorem 20** *Let $U \subseteq V \subseteq K$ and $\phi(x_V) = \phi(i_V, y_V) = \phi(i_U, i_{V\setminus U}, y_V)$ be a CG-potential defined on $\mathcal{X}_V = \mathcal{I}_V \times \mathcal{Y}_V = (\mathcal{I}_U \times \mathcal{I}_{V\setminus U}) \times \mathcal{Y}_V$ with canonical characteristics $(g, h, J)$, and moment characteristics $(p, \xi, \Sigma)$. Let $\eta(x_U) = \eta(i_U, y_V) = \sum_{i_{V\setminus U}} \phi(x_V)$ be the marginal of $\phi(x_V)$ with respect to $I_{V\setminus U}$. Let $\phi'(x_U) = \phi'(i_U, y_V)$ be a CG-potential with moment characteristics $(p', \xi', \Sigma')$ where:*

$$
\begin{aligned}
p'(i_U) &= \sum_{i_{V\setminus U}} p(i_U, i_{V\setminus U}) \\
\xi'(i_U) &= \sum_{i_{V\setminus U}} \xi(i_U, i_{V\setminus U}) p(i_U, i_{V\setminus U}) / p'(i_U) \\
\Sigma'(i_U) &= \sum_{i_{V\setminus U}} \Sigma(i_U, i_{V\setminus U}) p(i_U, i_{V\setminus U}) / p'(i_U) \\
&\quad + \sum_{i_{V\setminus U}} (\xi(i_U, i_{V\setminus U}) - \xi(i_U))^T (\xi(i_U, i_{V\setminus U}) - \xi(i_U)) p(i_U, i_{V\setminus U}) / p'(i_U)
\end{aligned}
\tag{3.19}
$$

*Then if $h$ and $J$ depend on $i_{V\setminus U}$ the marginal $\eta(x_U)$ will* **not** *be a CG-potential but it will possess the same moment characteristics as the CG-potential $\phi'(x_U)$.*

**Proof.**

The potential $\phi(i_U, i_{V\setminus U}, y_V)$ expressed in terms of its moment characteristics $(p, \xi, \Sigma)$ may be written as follows:

$$
\begin{aligned}
\phi(i_U, i_{V\setminus U}, y_V) &= \chi(i_U, i_{V\setminus U}) p(i_U, i_{V\setminus U}) \{\det\Sigma(i_U, i_{V\setminus U})\}^{-1/2} (2\Pi)^{-1/2|\Gamma_V|} \\
&\quad \times \exp\left\{-\frac{1}{2}(y_V - \xi(i_U, i_{V\setminus U}))^T \Sigma(i_U, i_{V\setminus U})^{-1} (y_V - \xi(i_U, i_{V\setminus U}))\right\}
\end{aligned}
\tag{3.20}
$$

This may be carried out using the conversion formulae of Equations 3.5. Summing $\phi(i_U, i_{V\setminus U}, y_V)$ over $i_{V\setminus U}$ we obtain a new potential $\eta(i_U, y_V)$ where:

$$
\begin{aligned}
\eta(i_U, y_V) &= \sum_{i_{V\setminus U}} \phi(i_U, i_{V\setminus U}, y_V) \\
&= \sum_{i_{V\setminus U}} \Bigg( \chi(i_U, i_{V\setminus U}) p(i_U, i_{V\setminus U}) \{\det\Sigma(i_U, i_{V\setminus U})\}^{-1/2} (2\Pi)^{-1/2|\Gamma_V|} \\
&\quad \times \exp\left\{-\frac{1}{2}(y_V - \xi(i_U, i_{V\setminus U}))^T \Sigma(i_U, i_{V\setminus U})^{-1} (y_V - \xi(i_U, i_{V\setminus U}))\right\} \Bigg)
\end{aligned}
\tag{3.21}
$$

Since $h$ and $J$ depend on $i_{V\setminus U}$ then, by Equations 3.5, $\xi$ and $\Sigma$ will depend on $i_{V\setminus U}$ also. We are therefore unable to manipulate Equation 3.21 so that $\eta(i_U, y_V)$ forms a CG-potential. Instead $\eta(i_U, y_V)$ is a mixture of CG-potentials - we will term this a *CGM-potential.*

Let us define the moment characteristics of $\eta(i_U, y_V)$ to be $(p', \xi', \Sigma')$ where:

$$p'(i_U) = P(I_U = i_U),\ \xi'(i_U) = E[Y_V \mid I_U = i_U],\ \Sigma'(i_U) = V(Y_V \mid I_U = i_U)$$

then we may uniquely determine them as follows:

$$\begin{aligned}
p'(i_U) &= P(I_U = i_U) \\
&= \sum_{i_{V\setminus U}} p(i_U, i_{V\setminus U}) \\
\xi'(i_U) &= E[Y_V \mid I_U = i_U] \\
&= E\left[E\left[Y_V \mid (I_U, I_{V\setminus U})\right] \mid I_U = i_U\right] \\
&= E\left[\xi(I_U, i_{V\setminus U}) \mid I_U = i_U\right] \\
&= \sum_{i_{V\setminus U}} \xi(i_U, i_{V\setminus U}) p(i_U, i_{V\setminus U}) / \sum_{i_{V\setminus U}} p(i_U, i_{V\setminus U}) \\
&= \sum_{i_{V\setminus U}} \xi(i_U, i_{V\setminus U}) p(i_U, i_{V\setminus U}) / p'(i_U) \\
\Sigma'(i_U) &= V(Y_V \mid I_U = i_U) \\
&= E\left[V(Y_V \mid (I_U, I_{V\setminus U})) \mid I_U = i_U\right] + V\left(E\left[Y_V \mid (I_U, I_{V\setminus U})\right] \mid I_U = i_U\right) \\
&= E\left[\Sigma(I_U, i_{V\setminus U}) \mid I_U = i_U\right] + V\left(\xi(I_U, i_{V\setminus U}) \mid I_U = i_U\right) \\
&= E\left[\Sigma(I_U, i_{V\setminus U}) \mid I_U = i_U\right] \\
&+ E\Big[\Big(\xi(i_U, i_{V\setminus U}) - E\left[\xi(I_U, i_{V\setminus U}) \mid I_U = i_U\right]\Big)^T \\
&\times \Big(\xi(i_U, i_{V\setminus U}) - E\left[\xi(I_U, i_{V\setminus U}) \mid I_U = i_U\right]\Big)\Big] \\
&= E\left[\Sigma(I_U, i_{V\setminus U}) \mid I_U = i_U\right] \\
&+ E\left[\Big(\xi(i_U, i_{V\setminus U}) - \xi'(i_U)\Big)^T \Big(\xi(i_U, i_{V\setminus U}) - \xi'(i_U)\Big)\right] \\
&= \sum_{i_{V\setminus U}} \Sigma(i_U, i_{V\setminus U}) p(i_U, i_{V\setminus U}) / p'(i_U) \\
&+ \sum_{i_{V\setminus U}} \Big(\xi(i_U, i_{V\setminus U}) - \xi'(i_U)\Big)^T \Big(\xi(i_U, i_{V\setminus U}) - \xi'(i_U)\Big) p(i_U, i_{V\setminus U}) / p'(i_U)
\end{aligned}$$

So $\eta(i_U, y_V)$ has the same moment characteristics as $\phi'(i_U, y_V)$ as defined in Equations 3.19. $\phi'(i_U, y_V)$ will additionally require a zero/one dummy $\chi'(i_U)$ which is as defined in Equation 3.18.

□

### 3.12.5 Marginalising over Continuous Variables

**Definition 34 Continuous Marginalisation of a CG-Potential:** *Let* $U \subseteq V \subseteq K$ *and let* $\phi(x_V) = \phi(i_V, y_V) = \phi(i_V, y_{V\setminus U}, y_U)$ *be a CG-potential defined on* $\mathcal{X}_V = \mathcal{I}_V \times \mathcal{Y}_V = \mathcal{I}_V \times (\mathcal{Y}_{V\setminus U} \times \mathcal{Y}_U)$. *Then we will let the expression* $\sum_{V\setminus U} \phi(x_V)$ *denote the* **marginalisation** *of* $\phi(x_V)$, *with respect to* $Y_{V\setminus U}$, *to a function* $\eta(x_U)$ *defined on the space* $\mathcal{X}_U = \mathcal{I}_V \times \mathcal{Y}_U$ *where:*

$$\eta(x_U) = \sum_{V\setminus U} \phi(x_V) = \int_{y_{V\setminus U}=-\infty}^{y_{V\setminus U}=+\infty} \phi(x_V)\, dy_{V\setminus U} \tag{3.22}$$

**Theorem 21** *Let* $U \subseteq V \subseteq K$ *and* $\phi(x_V) = \phi(i_V, y_V) = \phi(i_V, y_{V\setminus U}, y_U)$ *be a CG-potential defined on* $\mathcal{X}_V = \mathcal{I}_V \times \mathcal{Y}_V = \mathcal{I}_V \times (\mathcal{Y}_{V\setminus U} \times \mathcal{Y}_U)$ *with canonical characteristics* $(g, h, J)$ *and let:*

$$y_V = \begin{pmatrix} y_{V\setminus U} \\ y_U \end{pmatrix} \; h(i_V) = \begin{pmatrix} h_{V\setminus U}(i_V) \\ h_U(i_V) \end{pmatrix} \; J(i_V) = \begin{pmatrix} J_{V\setminus U, V\setminus U}(i_V) & J_{V\setminus U, U}(i_V) \\ J_{U, V\setminus U}(i_V) & J_{U,U}(i_V) \end{pmatrix}$$

*Then the marginalisation of* $\phi(x_V)$ *with respect to* $Y_{V\setminus U}$ *is a CG-potential* $\eta(x_U) = \eta(i_V, y_U)$, *say, with canonical characteristics* $(g', h', J')$ *where:*

$$\begin{aligned} g'(i_V) &= g(i_V) + \frac{1}{2}\Big\{ |\, \Gamma_{V\setminus U} \,|\; log(2\Pi) - log\; det\, J_{V\setminus U, V\setminus U}(i_V) \\ &+ h_{V\setminus U}(i_V)^T J_{V\setminus U, V\setminus U}(i_V)^{-1} h_{V\setminus U}(i_V) \Big\} \\ h'(i_V) &= h_U(i_V) - J_{U, V\setminus U}(i_V) J_{V\setminus U, V\setminus U}(i_V)^{-1} h_{V\setminus U}(i_V) \\ J'(i_V) &= J_{U,U}(i_V) - J_{U, V\setminus U}(i_V) J_{V\setminus U, V\setminus U}(i_V)^{-1} J_{V\setminus U, U}(i_V) \end{aligned} \tag{3.23}$$

**Proof.**

We may express $\phi(i_V, y_{V\setminus U}, y_U)$ as follows:

$$\begin{aligned} &\phi(i_V, y_{V\setminus U}, y_U) \\ &= \chi(i_V) \exp\Big\{ g(i_V) + h(i_V)^T y_V - \frac{1}{2} y_V^T J(i_V) y_V \Big\} \\ &= \chi(i_V) \exp\Big\{ g(i_V) + \begin{pmatrix} h_{V\setminus U}(i_V)^T & h_U(i_V)^T \end{pmatrix} \begin{pmatrix} y_{V\setminus U} \\ y_U \end{pmatrix} \\ &\quad - \frac{1}{2} \begin{pmatrix} y_{V\setminus U}^T & y_U^T \end{pmatrix} \begin{pmatrix} J_{V\setminus U, V\setminus U}(i_V) & J_{V\setminus U, U}(i_V) \\ J_{U, V\setminus U}(i_V) & J_{U,U}(i_V) \end{pmatrix} \begin{pmatrix} y_{V\setminus U} \\ y_U \end{pmatrix} \Big\} \end{aligned}$$

$$
\begin{aligned}
&= \chi(iv)\exp\Big\{g(iv) + h_{V\setminus U}(iv)^T y_{V\setminus U} + h_U(iv)^T y_U \\
&\quad - \frac{1}{2}\Big(y_{V\setminus U}^T J_{V\setminus U,V\setminus U}(iv) y_{V\setminus U} + y_{V\setminus U}^T J_{V\setminus U,U}(iv) y_U \\
&\quad + y_U^T J_{U,V\setminus U}(iv) y_{V\setminus U} + y_U^T J_{U,U}(iv) y_U\Big)\Big\} \\
&= \chi(iv)\exp\Big\{-\frac{1}{2}\Big(y_{V\setminus U}^T J_{V\setminus U,V\setminus U}(iv) y_{V\setminus U} + y_{V\setminus U}^T J_{V\setminus U,U}(iv) y_U \\
&\quad + y_U^T J_{U,V\setminus U}(iv) y_{V\setminus U} - 2h_{V\setminus U}(iv)^T y_{V\setminus U}\Big) \\
&\quad + g(iv) + h_U(iv)^T y_U - \frac{1}{2} y_U^T J_{U,U}(iv) y_U\Big\} \\
&= \chi(iv)\exp\Big\{-\frac{1}{2}\Big(y_{V\setminus U}^T + y_U^T J_{U,V\setminus U}(iv) J_{V\setminus U,V\setminus U}(iv)^{-1} \\
&\quad - h_{V\setminus U}(iv)^T J_{V\setminus U,V\setminus U}(iv)^{-1}\Big) J_{V\setminus U,V\setminus U}(iv) \\
&\quad \Big(y_{V\setminus U} + J_{V\setminus U,V\setminus U}(iv)^{-1} J_{V\setminus U,U}(iv) y_U - J_{V\setminus U,V\setminus U}(iv)^{-1} h_{V\setminus U}(iv)\Big) \\
&\quad + \frac{1}{2}\Big(y_U^T J_{U,V\setminus U}(iv) J_{V\setminus U,V\setminus U}(iv)^{-1} J_{V\setminus U,U}(iv) y_U \\
&\quad - y_U^T J_{U,V\setminus U}(iv) J_{V\setminus U,V\setminus U}(iv)^{-1} h_{V\setminus U}(iv) \\
&\quad - h_{V\setminus U}(iv)^T J_{V\setminus U,V\setminus U}(iv)^{-1} J_{V\setminus U,U}(iv) y_U \\
&\quad + h_{V\setminus U}(iv)^T J_{V\setminus U,V\setminus U}(iv)^{-1} h_{V\setminus U}(iv)\Big) \\
&\quad + g(iv) + h_U(iv)^T y_U - \frac{1}{2} y_U^T J_{U,U}(iv) y_U\Big\} \\
&= \chi(iv)\exp\Big\{-\frac{1}{2}\Big(y_{V\setminus U} + J_{V\setminus U,V\setminus U}(iv)^{-1} J_{V\setminus U,U}(iv) y_U \\
&\quad - J_{V\setminus U,V\setminus U}(iv)^{-1} h_{V\setminus U}(iv)\Big)^T J_{V\setminus U,V\setminus U}(iv) \\
&\quad \Big(y_{V\setminus U} + J_{V\setminus U,V\setminus U}(iv)^{-1} J_{V\setminus U,U}(iv) y_U - J_{V\setminus U,V\setminus U}(iv)^{-1} h_{V\setminus U}(iv)\Big) \\
&\quad + g(iv) + \frac{1}{2} h_{V\setminus U}(iv)^T J_{V\setminus U,V\setminus U}(iv)^{-1} h_{V\setminus U}(iv) \\
&\quad + \Big(h_U(iv) - J_{U,V\setminus U}(iv) J_{V\setminus U,V\setminus U}(iv)^{-1} h_{V\setminus U}(iv)\Big)^T y_U \\
&\quad - \frac{1}{2} y_U^T \Big(J_{U,U}(iv) - J_{U,V\setminus U}(iv) J_{V\setminus U,V\setminus U}(iv)^{-1} J_{V\setminus U,U}(iv)\Big) y_U\Big\}
\end{aligned}
\tag{3.24}
$$

Thus using Equation 3.24:

$$
\begin{aligned}
&\int_{y_{V\setminus U}=-\infty}^{y_{V\setminus U}=+\infty} \phi(iv, y_{V\setminus U}, y_U)\, dy_{V\setminus U} \\
&\quad = \chi(iv)\exp\Big\{g(iv) + \frac{1}{2} h_{V\setminus U}(iv)^T J_{V\setminus U,V\setminus U}(iv)^{-1} h_{V\setminus U}(iv) \\
&\qquad + \Big(h_U(iv) - J_{U,V\setminus U}(iv) J_{V\setminus U,V\setminus U}(iv)^{-1} h_{V\setminus U}(iv)\Big)^T y_U
\end{aligned}
$$

$$- \frac{1}{2} y_U^T \left( J_{U,U}(iv) - J_{U,V\setminus U}(iv) J_{V\setminus U,V\setminus U}(iv)^{-1} J_{V\setminus U,U}(iv) \right) y_U \Big\}$$
$$\times \int_{y_{V\setminus U}=-\infty}^{y_{V\setminus U}=+\infty} \exp\Big\{ - \frac{1}{2} \Big( y_{V\setminus U} + J_{V\setminus U,V\setminus U}(iv)^{-1} J_{V\setminus U,U}(iv) y_U$$
$$- J_{V\setminus U,V\setminus U}(iv)^{-1} h_{V\setminus U}(iv) \Big)^T J_{V\setminus U,V\setminus U}(iv) \Big( y_{V\setminus U}$$
$$+ J_{V\setminus U,V\setminus U}(iv)^{-1} J_{V\setminus U,U}(iv) y_U - J_{V\setminus U,V\setminus U}(iv)^{-1} h_{V\setminus U}(iv) \Big) \Big\} dy_{V\setminus U} \tag{3.25}$$

But since:

$$\int_{y_A=-\infty}^{y_A=+\infty} \exp\Big\{ - \frac{1}{2} \big( y_A - \mu_A \big)^T J \big( y_A - \mu_A \big) \Big\} dy_A = (2\Pi)^{|\Gamma_A|/2} (\det J)^{-1/2}$$

Equation 3.25 gives:

$$\int_{y_{V\setminus U}=-\infty}^{y_{V\setminus U}=+\infty} \phi(iv, y_{V\setminus U}, y_U)\, dy_{V\setminus U}$$
$$= (2\Pi)^{|\Gamma_{V\setminus U}|/2} \Big( \det J_{V\setminus U,V\setminus U}(iv) \Big)^{-1/2} \chi(iv) \exp\Big\{ g(iv)$$
$$+ \frac{1}{2} h_{V\setminus U}(iv)^T J_{V\setminus U,V\setminus U}(iv)^{-1} h_{V\setminus U}(iv)$$
$$+ \Big( h_U(iv) - J_{U,V\setminus U}(iv) J_{V\setminus U,V\setminus U}(iv)^{-1} h_{V\setminus U}(iv) \Big)^T y_U$$
$$- \frac{1}{2} y_U^T \Big( J_{U,U}(iv) - J_{U,V\setminus U}(iv) J_{V\setminus U,V\setminus U}(iv)^{-1} J_{V\setminus U,U}(iv) \Big) y_U \Big\}$$
$$= \chi(iv) \exp\Big\{ \Big( g(iv) + \frac{1}{2} \big( |\, \Gamma_{V\setminus U} \,| \log(2\Pi) - \log\det J_{V\setminus U,V\setminus U}(iv)$$
$$+ h_{V\setminus U}(iv)^T J_{V\setminus U,V\setminus U}(iv)^{-1} h_{V\setminus U}(iv) \big) \Big)$$
$$+ \Big( h_U(iv) - J_{U,V\setminus U}(iv) J_{V\setminus U,V\setminus U}(iv)^{-1} h_{V\setminus U}(iv) \Big)^T y_U$$
$$- \frac{1}{2} y_U^T \Big( J_{U,U}(iv) - J_{U,V\setminus U}(iv) J_{V\setminus U,V\setminus U}(iv)^{-1} J_{V\setminus U,U}(iv) \Big) y_U \Big\}$$
$$= \chi(iv) \exp\Big\{ g'(iv) + h'(iv)^T y_U - y_U^T J'(iv) y_U \Big\}$$

where $(g', h', J')$ are the canonical characteristics given in Equations 3.23.

□

In our *Mathematica* implementation if we are required to marginalise over a set of discrete variables and a set of continuous variables then we simply marginalise

over the set of continuous variables first and then over the set of discrete variables. The order in which the marginalisation is carried out is not important but marginalising over the continuous variables first reduces the dimension of the canonical characteristics. It is then simpler and hence faster to both determine which discrete marginalisation strategy is required and apply it. We thus have a marginalisation routine which applies the three different marginalisation strategies as required.

To distinguish between the various forms of marginalisation consider the marginalisation of a CG-potential $\phi_V$ with respect to a set of variables $X_{V\setminus U} = (I_{V\setminus U}, Y_{V\setminus U})$ denoted:

$$\sum_{V\setminus U} \phi_V$$

If we marginalise over the set of continuous variables, $Y_{V\setminus U}$, first and find that the resulting potential has canonical characteristics $(g, h, J)$, say, with $h$ and $J$ independent of $i_{V\setminus U}$ then we may apply Equations 3.17 to complete the marginalisation. In line with Lauritzen (1992) we term this marginalisation a *strong marginalisation*. Alternatively suppose that having marginalised with respect to the continuous variables, $Y_{V\setminus U}$, we find that the resulting potential has canonical characteristics $(g, h, J)$, say, with $h$ and $J$ dependent on $i_{V\setminus U}$. We would then apply Equations 3.19 to complete the marginalisation. We term this a *weak marginalisation*.

## 3.13 Propagation

We now consider the formation of a propagation algorithm based on the basic operations defined in Section 3.12.

### 3.13.1 Initialising the System

We initialise the junction tree in the usual way. First we let $a_C \equiv 1$ for all the cliques $C \in \mathcal{C}$, so the canonical characteristics are $(\{0\}, \{0\}, \{0\})$ where each $\{0\}$ represents a scalar, vector or matrix of zeros of appropriate dimension. Similarly we let $b_S \equiv 1$ for all the separators $S \in \mathcal{S}$.

We have already defined the conditional distribution of each variable in the system as a CG-potential, and have shown how the canonical characteristics of

such variables may be determined. We now wish to enter this information into the junction tree to set up our system.

For each variable $X_a \in X$, we assign $X_a$ to a clique, $C \in \mathcal{C}$. We may only assign $X_a$ to a clique which contains all the variables necessary to define the conditional distribution of $X_a \mid X_{pa(a)}$. If more than one suitable clique exists then the assignment of $X_a$ to any one of them is entirely arbitrary. The CG-potential of $X_a \mid X_{pa(a)}$, denoted $\phi_{a,pa(a)}$, is extended to occupy the same space as the clique $C$ and the CG-potential, $a_C$, of clique $C$ is then multiplied by this extension.

For the waste incinerator example we assign $B$ and $C$ to $\{C, B\}$, $D$ to $\{D, E, B, W\}$, $E$, $F$ and $W$ to $\{E, B, F, W\}$, $L$ to $\{D, L\}$, $Mi$ to $\{D, Mi, W\}$ and $M0$ to $\{M0, D, Mi\}$.

In our implementation we define an assignment of variables to cliques. This assignment is a list of lists of variables that correspond positionally to the cliques in the list `cliques`. Our assignment for the waste incinerator example is as follows:

```
assignment = { {c, b}, {e, f, w}, {d}, {mi}, {l}, {m0} }
```

Using this information and the previously defined conditional variables we are able to form the initial clique potentials, `cpots`, and initial separator potentials, `spots`. These are simply lists of CG-potentials which are positioned according to the positions of the corresponding cliques and separators in `cliques` and `seps`.

With the system defined as above and interpreting each $a_C$ as a CG-potential, the joint density $f_K$ of the random variables $X$ may, by virtue of the factorisation criterion, be written in the form:

$$f_K(x) = \prod_{C \in \mathcal{C}} a_C(x_C) \tag{3.26}$$

With our separator potentials defined as $b_S \equiv 1$ we may trivially write:

$$f_K(x) = \frac{\prod_{C \in \mathcal{C}} a_C(x_C)}{\prod_{S \in \mathcal{S}} b_S(x_C)} \tag{3.27}$$

Equation 3.27 may be written more generally as:

$$f = \frac{\prod_{C \in \mathcal{C}} a_C}{\prod_{S \in \mathcal{S}} b_S} \tag{3.28}$$

where we do not restrict $f$ to be a density. $f$ is termed the *joint system belief*. As in Section 2.12 we will term any collection of non-negative functions $\mathcal{K} = (\{a_C : C \in \mathcal{C}\}, \{b_S : S \in \mathcal{S}\})$ on the various cliques and separators a *charge* on $\mathcal{T}$.

**Theorem 22** *Following initialisation the joint system belief $f$ of the random variables $X$ is a CG-potential.*

**Proof.** Recall from Theorem 17 that the product of two CG-potentials is a CG-potential. Since each conditional distribution in the system is defined to be a CG-potential then their product, $f$, must be a CG-potential also.

□

### 3.13.2 Propagating a Schedule

We have created a tree which contains all the information we have on our variables and which has the correct joint system belief. We do not, necessarily, have the correct probabilities or moments for any variable in every clique of the tree containing it, since the information on a variable is entered into only one clique. Similarly no separator will contain the correct probabilities or moments for any variable it contains since the separator potentials were defined to be unitary. We therefore wish to make the cliques and separators *consistent* with each other so that they contain the same probability and moment information on the variables. The basic operation to obtain this is a flow (or *sum-flow*) as defined in Definition 23. Each flow affects the potentials on exactly one clique and one separator and is composed of the basic operations which were defined in Section 3.12. Lemma 4 which states that "the passage of a flow does not affect the contraction of a charge" continues to hold under our scheme.

The propagation scheme to make all cliques and separators consistent consists of two phases both of which are composed of flows only. For this particular propagation scheme to operate correctly a strong root is required. Recall from Theorem 13 that since the junction tree $\mathcal{T}$ was derived from a strongly decomposable graph it will contain at least one strong root and thus this criterion is satisfied. The first phase is termed the *collect evidence phase*. In this phase active flows are passed along edges which are directed towards a strong root $R$. The second phase

is termed the *distribute evidence phase*. In this phase active flows starting from $R$ are passed back towards the periphery.

**Proposition 8** *For any connected junction tree $\mathcal{T}$ with a strong root $R$, there exists a fully active schedule such that active flows may be directed towards $R$ in the collect evidence phase and passed back from $R$ towards the periphery in the distribute evidence phase.*

**Proof.** Recall the algorithm described in Proposition 6. This began with a connected tree $\mathcal{T}$ and deleted leaves from the tree to form a single clique $C^*$, say. At each stage of the algorithm two active flows were added to a propagation schedule resulting in a fully active schedule. This algorithm may be rewritten such that it begins with a tree composed of a single clique $C^*$ and adds leaves to it until it forms $\mathcal{T}$. Again, two active flows are added at each stage of the algorithm. This algorithm is as follows:

If $\mathcal{T}$ consists of a single clique $C^*$ then the result is vacuously true. Otherwise, let $\mathcal{T}$ be a connected tree with $n$ separators. We may then form a fully active schedule for $\mathcal{T}$ if we apply the following strategy:

(i) Start with an empty schedule and a tree $\mathcal{T}_m$. Let $m = 1$ and $\mathcal{T}_1 = C^*$.

(ii) Arbitrarily select a leaf, $C_{0,m}$, from $\mathcal{T}_m$ such that there exists a clique, $C_{1,m}$, in $\mathcal{T} \backslash \mathcal{T}_m$ joined to $C_{0,m}$ by a separator $S_{0,m}$.

(iii) Let the $(n+1-m)$-th flow in the schedule be from $C_{1,m}$ to $C_{0,m}$ via $S_{0,m}$. Let the $(n+m)$-th flow be from $C_{0,m}$ to $C_{1,m}$ via $S_{0,m}$.

(iv) Add $C_{1,m}$ and $S_{0,m}$ to $\mathcal{T}_m$ to form $\mathcal{T}_{m,0}$.

(v) If $\mathcal{T}_{m,0} = \mathcal{T}$ then end. Otherwise let $m = m+1$, $\mathcal{T}_m = \mathcal{T}_{m-1,0}$ and repeat from step (ii).

If we let $C^* = R$ then we have the required schedule.

□

In our *Mathematica* implementation we define a *propagation schedule* according Proposition 8. This schedule is simply a list of which flows should occur and in what order. Each flow is a list of the source, the separator, and the sink. It contains all the information necessary to define the physical structure of the junction tree.

A suitable propagation schedule, sched, for the waste incinerator example is as follows:

```
sched = {{{{m0,d,mi},{}},  {{d,mi},{},5},  {{d,mi},{w}}},
         {{{d,mi},{w}},    {{d},{w},3},     {{d,e},{b,w}}},
         {{{d,l},{}},      {{d},{},4},      {{d,e},{b,w}}},
         {{{d,e},{b,w}},   {{e},{b,w},2},   {{e},{b,f,w}}},
         {{{e},{b,f,w}},   {{},{b},1},      {{c},{b}}},
         {{{c},{b}},       {{},{b},1},      {{e},{b,f,w}}},
         {{{e},{b,f,w}},   {{e},{b,w},2},   {{d,e},{b,w}}},
         {{{d,e},{b,w}},   {{d},{},4},      {{d,l},{}}},
         {{{d,e},{b,w}},   {{d},{w},3},     {{d,mi},{w}}},
         {{{d,mi},{w}},    {{d,mi},{},5},   {{m0,d,mi},{}}}}}
```

**Theorem 23** *Suppose that active flows directed towards a strong root $R$ are passed along the edges of an initialised junction tree $\mathcal{T}$ in the collect evidence phase of a fully active propagation schedule then every marginalisation required will be strong.*

**Proof.** Recall the definition of a strong root. A clique $R$ is said to be a strong root if any pair $A$, $B$ of neighbouring cliques on $\mathcal{T}$, with $A$ the closer of the pair to $R$, satisfies:

$$(B \backslash A) \subseteq \Gamma \vee (B \cap A) \subseteq \Delta$$

By definition, any active flow in the collect evidence phase will be passed between a pair of neighbouring cliques $A$ and $B$ such that $A$ is closer to $R$ than $B$. By Equations 2.5 the updated separator potential $b^*_{A \cap B}$ resulting from this flow is formed by marginalising $a_B$, the clique potential for $B$ , with respect to $B \backslash A$. If $(B \backslash A) \subseteq \Gamma$ then the marginalisation will be over continuous variables only, and hence strong. Alternatively if $(B \cap A) \subseteq \Delta$ then we may marginalise $a_B$ over any continuous variables, $Y_B$, it contains. This will ensure that the resulting potential, with canonical characteristics $(g, h, J)$, say, contains only discrete variables and hence $h = J = 0$. The marginalisation with respect to $(B \backslash A) \cap \Delta$ will therefore be strong also.

□

A consequence of Theorem 23 is that, following the collect evidence phase, the potential $b_{B \cap A}$ on every separator $B \cap A$ between neighbouring cliques $A$ and

$B$, with potentials $a_A$ and $a_B$ respectively, is the strong marginal of $a_B$ if $A$ is the closer of the pair to $R$. Moreover, the potentials $a_A$, $a_B$, and $b_{B\cap A}$ are all ensured to be CG-potentials. Also, since the collect evidence phase consists only of strong marginalisations then, during this phase, Theorem 14 and Corollaries 1 and 2 continue to hold.

**Theorem 24** *The CG-potential of the strong root $R$ will be $f_R$ following the collect evidence phase.*

**Proof.** Following the collect evidence phase $R$ will have received active flows from each of its neighbours and will hence be live. By Corollary 2 its potential is thus $f_R$.

□

**Lemma 5** *Let $A$ and $B$ be adjacent cliques in a junction tree $\mathcal{T}$ with $A$ the closer of the pair to a strong root $R$. Let $S = A \cap B$ be the separator which joins $A$ and $B$. Let $a_B$ and $b_S$ be functions on $B$ and $S$ respectively then:*

$$\sum_{B\backslash A} a_B b_S = b_S \sum_{B\backslash A} a_B$$

**Proof.** Suppose that $(B\backslash A) \subseteq \Gamma$ then:

$$\sum_{B\backslash A} a_B b_S = \int_{y_{B\backslash A}} a_B b_S dy_{B\backslash A} = b_S \int_{y_{B\backslash A}} a_B dy_{B\backslash A} = b_S \sum_{B\backslash A} a_B$$

Otherwise $(B \cap A) \subseteq \Delta$ and so we may first integrate over all the continuous variables $Y_{B\backslash A} = Y_B$ to obtain a discrete potential $a'_B b_S$ where:

$$a'_B = \int_{y_B} a_B dy_B$$

We may then sum over the variables $I_{B\backslash A}$. This will be a strong marginalisation so:

$$\sum_{i_{B\backslash A}} a'_B b_S = b_S \sum_{i_{B\backslash A}} a'_B$$

Hence the result.

□

**Lemma 6** *Let $A$ and $B$ be adjacent cliques in a junction tree $\mathcal{T}$ with $A$ the closer of the pair to a strong root $R$. Let $S = A \cap B$ be the separator which joins $A$ and $B$. Let $a_A$, $a_B$ and $b_S$ be the CG-potentials of $A$, $B$ and $S$, respectively. If a flow is passed from $A$ to $B$ via $S$ and $b_S$ is the strong marginal of $a_B$, then $A$ and $B$ are calibrated after the passage of the flow:*

$$\sum_{B\setminus A} a'_B = b'_S = \sum_{A\setminus B} a'_A$$

*Where $a'_A$, $a'_B$ and $b'_S$ are the updated CG-potentials of $A$, $B$ and $S$.*

**Proof.**

Using Equations 2.5 and noting that $a'_A = a_A$ we find that:

$$b'_S = \sum_{A\setminus B} a_A = \sum_{A\setminus B} a'_A$$

and:

$$\sum_{B\setminus A} a'_B = \sum_{B\setminus A} a_B b^*_S$$

where $b^*_S = \sum_{A\setminus B} a'_A / \sum_{B\setminus A} a_B$. But by Lemma 5 $\sum_{B\setminus A} a_B b^*_S = b^*_S \sum_{B\setminus A} a_B$ so:

$$\sum_{B\setminus A} a'_B = \left(\frac{\sum_{A\setminus B} a'_A}{\sum_{B\setminus A} a_B}\right) \sum_{B\setminus A} a_B = \sum_{A\setminus B} a'_A = b'_S$$

□

**Theorem 25** *Suppose a propagation schedule consisting of a collect evidence phase, in which flows are passed along edges which are directed towards a strong root $R$, and a distribute evidence phase, in which flows starting from $R$ are passed back towards the periphery, is passed. Then the resulting tree of belief universes is locally calibrated.*

**Proof.** Consider any pair of neighbouring cliques $A$ and $B$ with $A$ the closer of the pair to $R$. Let $S$ be the separator which joins them. After the collect evidence phase a flow will have been passed from $B$ to $A$. The CG-potential on $S$ will therefore be the strong marginal of the CG-potential on $B$. After the distribute evidence phase a flow will have been passed from $A$ to $B$. Thus by Lemma 6 $A$ and $B$ will be calibrated. Due to the arbitrary nature of the choice of $A$ and $B$ every pair of neighbouring cliques will then be calibrated. Hence the result.

□

By Theorem 15 if a flow is passed across a separator which joins two mutually calibrated cliques then the potentials of the cliques and separators are invariant to the flow. Thus when a full schedule of flows, defined as in Proposition 8, has been passed, every pair of neighbouring cliques will be calibrated and hence no additional flow will affect the potentials comprising the charge. The system will have thus reached equilibrium.

**Theorem 26** *Let $\mathcal{T}$ be a locally consistent junction tree with a strong root $R$ and $n$ cliques $\mathcal{C}$. Let $f_K$ be the joint system belief for $\mathcal{T}$ and let $C \in \mathcal{C}$. Then:*

$$\sum_{K \backslash C} f_K \propto a_C \tag{3.29}$$

**Proof.** The result is trivial if $n = 1$ in which case $K = C$. Suppose therefore that $n = 2$. Let $R$ be the strong root in $\mathcal{T}$ and $L$ be the other clique then $K = R \cup L$ and the separator joining $R$ to $L$ is $S = R \cap L$. By Lemma 6:

$$\sum_{K \backslash R} f_K = \sum_{L \backslash R} f_K = \sum_{L \backslash R} \frac{a_L a_R}{b_S} = \frac{a_R}{b_S} \sum_{L \backslash R} a_L = a_R$$

Thus Equation 3.29 is satisfied for $C = R$. Since $R$ is a strong root either $S \subseteq \Delta$ or $(L \backslash R) \subseteq \Gamma$. If $S \subseteq \Delta$ then:

$$\sum_{K \backslash L} f_K = \sum_{R \backslash L} f_K = \sum_{R \backslash L} \frac{a_L a_R}{b_S} = \frac{a_L}{b_S} \sum_{R \backslash L} a_R = a_L$$

and Equation 3.29 is satisfied for $C = L$. Alternatively suppose that $(L \backslash R) \subseteq \Gamma$ then there are only continuous variables in $L \backslash S$. Let $(i_S, y_S)$ denote the states of the variables in $S$ and $y_{L \backslash S}$ denote the states of the variables in $L \backslash S$. We note that since $b_S$ is the weak marginal of $a_R$ then $b_S$ is the weak marginal of $f_K$:

$$b_S = \sum_{R \backslash S} a_R = \sum_{R \backslash S} \sum_{K \backslash R} f_K = \sum_{K \backslash R} \sum_{R \backslash S} f_K = \sum_{K \backslash S} f_K$$

The moments $p(i_S)$, $E[Y_S \mid I_S = i_S]$ and $V(Y_S \mid I_S = i_S)$ are therefore correct when calculated according to $b_S$ or $a_L$ by construction of the marginalisation operators. The remaining moments, $E[Y_{L \backslash S} \mid I_S = i_S]$, $V(Y_{L \backslash S} \mid I_S = i_S)$ and $\mathrm{Cov}(Y_{L \backslash S}, Y_S \mid I_S = i_S)$ may, by definition of the conditional probability distributions $Y_{L \backslash S} \mid (I_S, Y_S)$, be expressed as:

$$
\begin{aligned}
E[Y_{L\setminus S} \mid I_S = i_S] &= A(i_S) + B(i_S)^T E[Y_S \mid I_S = i_S] \\
V(Y_{L\setminus S} \mid I_S = i_S) &= G(i_S) + B(i_S)^T V(Y_S \mid I_S = i_S) B(i_S) \\
\mathrm{Cov}(Y_{L\setminus S}, Y_S \mid I_S = i_S) &= B(i_S)^T V(Y_S \mid I_S = i_S)
\end{aligned}
$$

where $A(i_S)$, $B(i_S)$ and $G(i_S)$ are determined from $a_L/b_S$ alone. Therefore since $E[Y_S \mid I_S = i_S]$ and $V(Y_S \mid I_S = i_S)$ are correct then the remaining moments will be correct also. Thus $a_L$ is the correct weak marginal of $f_K$. Hence Equation 3.29 is satisfied for $C = L$.

Now suppose that $n > 2$. Select a leaf $L$ arbitrarily from $\mathcal{C}\setminus R$ and put $R' = \cup_{C \in \mathcal{C}\setminus\{L\}} C$. Then $R'$ is a strong root and we may apply the case where $n = 2$ to $L$ and $R'$. Selecting all possible leaves $L$ we obtain the result.

□

Thus our scheme is complete. We have described a propagation technique which ensures equality of moments in all the clique and separator CG-potentials. Since the junction tree is locally consistent we may calculate the moments of any variable by weakly marginalising any CG-potential. It should be noted that if $K = \Delta$ then we have the pure discrete case, similarly if $K = \Gamma$ we have the pure continuous case. In both these cases all marginalisations are ensured to be strong and we will be able to obtain the correct probability density functions of every variable.

In our *Mathematica* implementation we have a simple propagation routine which takes each item in the schedule in turn and runs a flow algorithm using the formulae in Equations 2.5. The results of applying this to routine to the waste incinerator example are given in Tables 3.6 and 3.7.

| **B: Burning Regime** | | **F: Filter State** | | **W: Type of Waste** | |
|---|---|---|---|---|---|
| $b$ | $P(B = b)$ | $f$ | $P(F = f)$ | $w$ | $P(W = w)$ |
| Stable | 0.850000 | Intact | 0.950000 | Industrial | 0.285714 |
| Unstable | 0.150000 | Defective | 0.050000 | Household | 0.714286 |

Table 3.6: The marginal distributions of the discrete variables in the waste incinerator problem.

| Variable | Mean | Variance |
|---|---|---|
| C: $CO_2$ in Emission | -1.850000 | 0.257500 |
| D: Emission of Dust | 3.039286 | 0.592909 |
| E: Filter Efficiency | -3.253571 | 0.502511 |
| L: Light Penetrability | 1.480357 | 0.398227 |
| Mi: Metal in Waste | -0.214286 | 0.210510 |
| M0: Emission of Metal | 2.825000 | 0.740113 |

Table 3.7: The moments of the marginal distributions of the continuous variables in the waste incinerator problem.

## 3.14 Entering Evidence

In this section we will consider how we may enter evidence into our system.

### 3.14.1 Entering Continuous Evidence

**Theorem 27** *Let $V \subseteq K$ be a universe in a junction tree $\mathcal{T}$. Let $\phi_V(x_V) = \phi_V(i_V, y_V)$, be the CG-potential for $V$ and suppose that it has canonical characteristics $(g, h, J)$. Let $Y_U$ denote a continuous variable in $Y_V$ and suppose we may partition $y_V$, $h$ and $J$ as follows:*

$$y_V = \begin{pmatrix} y_{V\setminus U} \\ y_U \end{pmatrix} \; h(i_V) = \begin{pmatrix} h_{V\setminus U}(i_V) \\ h_U(i_V) \end{pmatrix} \; J(i_V) = \begin{pmatrix} J_{V\setminus U, V\setminus U}(i_V) & J_{V\setminus U, U}(i_V) \\ J_{U, V\setminus U}(i_V) & J_{U,U}(i_V) \end{pmatrix} \tag{3.30}$$

*If the evidence $\mathcal{E} : Y_U = y_U^*$ is observed then, given this evidence, the updated potential $\phi'(x_V) = \phi'(i_V, y_U^*, y_{V\setminus U})$ on $V$ will be a CG-potential with canonical characteristics $(g', h', J')$ where:*

$$\begin{aligned} g'(i_V) &= g(i_V) + h_U(i_V) y_U^* + \frac{1}{2} J_{U,U}(i_V)(y_U^*)^2 \\ h'(i_V) &= h_{V\setminus U}(i_V) + J_{V\setminus U, U}(i_V) y_U^* \\ J'(i_V) &= J_{V\setminus U, V\setminus U}(i_V) \end{aligned} \tag{3.31}$$

**Proof.**

$$
\begin{aligned}
\phi'(x_V) &= \phi'(i_V, y^*_U, y_{V\setminus U}) \\
&= \chi(i_V)\exp\Big\{g(i_V) + h(i_V)^T y_V - \frac{1}{2} y_V^T J(i_V) y_V\Big\} \\
&= \chi(i_V)\exp\Big\{g(i_V) + \begin{pmatrix} h_{V\setminus U}(i_V)^T & h_U(i_V) \end{pmatrix}\begin{pmatrix} y_{V\setminus U} \\ y^*_U \end{pmatrix} \\
&- \frac{1}{2}\begin{pmatrix} y^T_{V\setminus U} & y^*_U \end{pmatrix}\begin{pmatrix} J_{V\setminus U, V\setminus U}(i_V) & J_{V\setminus U, U}(i_V) \\ J_{U, V\setminus U}(i_V) & J_{U,U}(i_V) \end{pmatrix}\begin{pmatrix} y_{V\setminus U} \\ y^*_U \end{pmatrix}\Big\} \\
&= \chi(i_V)\exp\Big\{g(i_V) + h_{V\setminus U}(i_V)^T y_{V\setminus U} + h_U(i_V) y^*_U \\
&- \frac{1}{2}\Big(y^T_{V\setminus U} J_{V\setminus U, V\setminus U}(i_V) y_{V\setminus U} + 2 y^*_U J_{U, V\setminus U}(i_V) y_{V\setminus U} + J_{U,U}(i_V)(y^*_U)^2\Big)\Big\} \\
&= \chi(i_V)\exp\Big\{\Big(g(i_V) + h_U(i_V) y^*_U - \frac{1}{2} J_{U,U}(i_V)(y^*_U)^2\Big) \\
&+ \Big(h_{V\setminus U}(i_V) - J_{V\setminus U, U}(i_V) y^*_U\Big)^T y_{V\setminus U} - \frac{1}{2} y^T_{V\setminus U} J_{V\setminus U, V\setminus U}(i_V) y_{V\setminus U}\Big\} \\
&= \chi(i_V)\exp\Big\{g'(i_V) + h'(i_V)^T y_{V\setminus U} - \frac{1}{2} y^T_{V\setminus U} J'(i_V) y_{V\setminus U}\Big\}
\end{aligned}
$$

Thus the canonical characteristics, $(g', h', J')$, are those given in Equation 3.31.

□

We may therefore enter evidence $y^*_U$ on a continuous variable $Y_U$ into a particular clique or separator by use of the formulae in Equations 3.31. Since these formulae alter the dimensions of $h$ and $J$ if we wish to enter continuous evidence in the entire system we must enter it into every clique and separator containing $Y_U$. When this has been completed the propagation schedule outlined in Section 3.13 may be used to update the system so that all the CG-potentials are consistent with this new information. The entry of continuous evidence affects each canonical characteristic $g$ linearly. A consequence of this is that the probabilities $p$ in any given universe may no longer sum to one. Therefore if we require our joint system belief to equal the joint probability distribution we must renormalise our system accordingly.

### 3.14.2 Entering Discrete Evidence

**Theorem 28** *Let $V \subseteq K$ be a universe in a junction tree $\mathcal{T}$. Let $\phi_V(x_V) = \phi_V(i_V, y_V)$, be the CG-potential for $V$ and let $I_U$ denote a discrete variable in $I_V$. If the evidence $\mathcal{E} : I_U = i^*_U$ is observed then, given this evidence, we may update the potential on $V$ by setting every $\chi(i_U, i_{V\setminus U}) = 0$ if $i_U \neq i^*_U$.*

**Proof.** A CG-potential for a universe $V$ is the product of an indicator variable $\chi(i_V)$ and a function of the canonical characteristics for $V$. Setting the indicator variable to zero for some level of $i_V$ will therefore zero out the entire potential for that level of $i_V$. Given the evidence that $I_U = i_U^*$ the potentials representing the probabilities that $I_U \neq i_U^*$ may thus be set to zero by setting $\chi(i_U, i_{V \setminus U}) = 0$ if $i_U \neq i_U^*$.

□

In order to enter discrete evidence $i_U^*$ on a variable $I_U$ we may proceed in one of two ways. The first method would be to enter the evidence through the indicator functions as described in Theorem 28. We may accomplish this by entering the evidence into just one clique which contains $I_U$. Propagating a schedule would then ensure that every indicator $\chi$, a function of $i_U$, is zero if $i_U^* \neq i_U$. The second method would be to remove all cells that have been deemed impossible by the evidence. This is equivalent to setting the unwanted $\chi$s to zero, however it reduces the size of the data structures which should increase the speed of computation. The drawback to this scheme is that we must apply the method to every universe containing $i_U$ before we propagate a schedule. Regardless of the chosen method, if we require our joint system belief to equal the joint probability distribution we must renormalise our system accordingly.

### 3.14.3 Evidence Entry in the Waste Incinerator Problem

In our *Mathematica* implementation of the scheme we enter discrete evidence using the second of the two methods. The addition of evidence on a single level of a discrete variable effectively reduces the dimension of any probability table containing it by one. Similarly the addition of evidence on a continuous variable will reduce the dimensions of any canonical characteristic formally a function of it. We therefore store lists, `fulldisvars` and `fullctsvars`, of the discrete and continuous variables we have evidence on so that we may take account of these changes in dimension. A list, `disdists`, of all the discrete variables and a list, `numdisvars`, of their dimensions is used to determine the number of cells in a particular universe. This latter list is updated when entering discrete evidence.

We enter two collections of evidence into the system. Both are standard examples considered elsewhere. We will use these examples later to compare the effectiveness of our different propagation techniques. Lauritzen (1992) considers

| B: Burning Regime | | F: Filter State | |
|---|---|---|---|
| $b$ | $P(B = b)$ | $f$ | $P(F = f)$ |
| Stable | 0.012253 | Intact | 0.999526 |
| Unstable | 0.987747 | Defective | 0.000474 |

| Variable | Mean | Variance |
|---|---|---|
| D: Emission of Dust | 3.607667 | 0.106179 |
| E: Filter Efficiency | -3.898338 | 0.005819 |
| Mi: Metal in Waste | 0.500000 | 0.010000 |
| M0: Emission of Metal | 4.107667 | 0.118179 |

Table 3.8: The probabilities, means and variances of the variables in the waste incinerator example given Lauritzen's evidence.

| B: Burning Regime | | F: Filter State | |
|---|---|---|---|
| $b$ | $P(B = b)$ | $f$ | $P(F = f)$ |
| Stable | 0.642434 | Intact | 0.785816 |
| Unstable | 0.357566 | Defective | 0.214184 |

| Variable | Mean | Variance |
|---|---|---|
| D: Emission of Dust | 3.774482 | 1.736158 |
| E: Filter Efficiency | -3.150352 | 2.061649 |
| Mi: Metal in Waste | 0.500000 | 0.010000 |
| M0: Emission of Metal | 4.274482 | 1.748158 |

Table 3.9: The probabilities, means and variances of the variables in the waste incinerator example given Olesen's evidence.

the case where it is known that industrial waste is being burned, light penetrability is 1.1 and the concentration of $CO_2$ is $-0.9$. The results of entering this information and repropagating the system are given in Table 3.8. Olesen (1991) considers the situation where it is known that industrial waste is being burned, light penetrability is 0.5, and the concentration of $CO_2$ is $-1.6$. The results for this collection of evidence are given in Table 3.9.

## 3.15 Simulation

We showed in Section 3.7.2 how in the purely discrete case, given possibly vacuous evidence $\mathcal{E}$, random realisations of the unobserved variables in the system may be simulated according to their joint conditional distribution given $\mathcal{E}$. We shall now consider how we might accomplish the same feat in the mixed case. With a slight change in notation to avoid confusion with the mean vectors $\xi$, we let $\zeta_A$ denote a set of simulated values of $X_A$ for a collection $A \subseteq K$. We shall let $X_E$, for $E \subseteq K$, denote the set of variables for which we have evidence and assume that we have the evidence $\mathcal{E} : X_E = x_E^*$.

First assume that the system has been initialised, evidence $\mathcal{E}$ has been entered, and a propagation schedule has not yet been passed. The representation $\mathcal{K}^*$ on the system now holds for a function $f^*$, where $f^*(x) = f(x\&\mathcal{E})$. We now apply the collect evidence phase of a propagation schedule to the system using sum-flows. Let us assume that the flows are directed towards some strong root $R$. The CG-potential $a_R(x_{R\setminus E}; x^*_{R\cap E})$, say, on clique $R$ is proportional to the distribution of the variables $X_{R\setminus E} = \{I_{R\setminus E}, Y_{R\setminus E}\}$ contained in $R$ given $\mathcal{E}$. We may use this CG-potential to simulate a set of values $\zeta_{R\setminus E} = (\zeta^I_{R\setminus E}, \zeta^Y_{R\setminus E})$ for the variables $X_{R\setminus E}$ in $R$ given $\mathcal{E}$.

We first simulate values for the discrete variables $I_{R\setminus E}$ in $R$. By determining the moment characteristics $(p, \xi, \Sigma)$ of $a_R$ we may produce a table of values $p'(i_{R\setminus E})$ such that $p'(i_{R\setminus E}) = p(i_{R\setminus E}) / \sum_{i_{R\setminus E}} p(i_{R\setminus E})$ represents the probability of obtaining a collection of discrete variables $i_{R\setminus E}$ given $\mathcal{E}$. Thus by forming cumulative values over the $p'(i_{R\setminus E})$ and simulating a single value, $u$, say, from a Uniform distribution $U \sim U(0,1)$ we may select the cell with the smallest cumulative value greater than $u$. This will determine our set of simulated values $\zeta^I_{R\setminus E}$ for the discrete variables $I_{R\setminus E}$ given the evidence $\mathcal{E}$. We then enter the evidence that $I_{R\setminus E} = \zeta^I_{R\setminus E}$ into the system. We now have the potential $a^I_R(y_{R\setminus E}; \zeta^I_{R\setminus E}, x^*_{R\cap E})$ on $R$.

In order to simulate the continuous variables in $R\backslash E$ we note that $a_R^I(y_{R\backslash E}; \zeta_{R\backslash E}^I, x_{R\cap E}^*)$ is proportional to a multivariate Normal distribution in $Y_{R\backslash E}$. In order to simulate a set of realisations $\zeta_{R\backslash E}^Y$ of the variables $Y_{R\backslash E}$ which is consistent with this multivariate Normal distribution we may simulate each variable $Y_a$, for $a \in \Gamma\cap (R\backslash E)$, in turn according to some appropriate univariate Normal distribution. The mean and variance of the marginal distribution of the first $Y_a$ may be determined from the moment characteristics of $a_R^I(y_{R\backslash E}; \zeta_{R\backslash E}^I, x_{R\cap E}^*)$. We may then simulate a value $\zeta_a$ of $Y_a$ using standard techniques. For example, we could simulate a pair of independent values $u_1$ and $u_2$ from a Uniform distribution $U \sim U(0,1)$ and using the polar-Marsaglian method transform $u_1$ and $u_2$ into two independent simulations from a standard Normal distribution. Picking one of these simulations we may scale it, using the mean and variance we calculated, to form $\zeta_a$. If we enter the evidence that $Y_a = \zeta_a$ into the system the potential on $R$ is now proportional to a multivariate Normal distribution in $Y_{R\backslash(a\cup E)}$. We can thus pick another continuous variable in $R$ simulate it using the updated potential on $R$ and enter it's simulated value as evidence into $R$. Repeating in this way we obtain the complete set of simulations $\zeta_{R\backslash E}^Y$, and hence $\zeta_{R\backslash E}$, for $R\backslash E$.

We may now pass a modified distribution phase of the propagation schedule. Consider a flow from a clique $A$ to a clique $B$ via a separator $S$. For the first flow of the distribution phase $A = R$. We will, for each flow, already have simulated a set of values $\zeta_{A\backslash E}$ for the variables $X_{A\backslash E}$ in $A$ given the evidence $\mathcal{E}$. Since the variables in $S$ form a subset of the variables in $A$ we may extract $\zeta_S$ directly from $\zeta_A$. Now pass the flow from $A$ to $B$ via $S$. Note that since $A$ contains evidence on all its variables the potential on $S$ equals the potential on $A$ and is a strong marginal. Thus the potential $a_B(x_{B\backslash(E\cup S)}; \zeta_{S\backslash E}, x_{B\cap E}^*)$, on $B$ is a CG-potential. We may simulate the remaining variables in $B$, discrete variables first, as outlined above for $R$. Proceeding in this way we may pass a full propagation schedule and generate a value for every variable from the desired joint distribution. If more than one simulation is required we store all variables formed after the collect evidence phase and repeat our simulation algorithm from this position for all subsequent simulations.

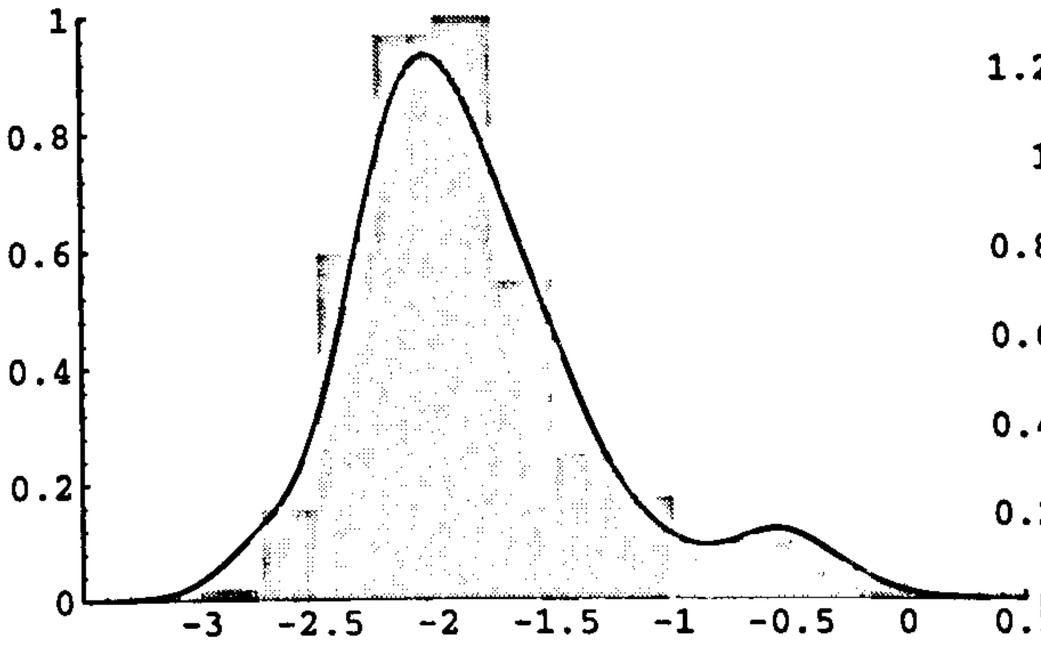
In our *Mathematica* implementation we used the simulation techniques described above to generate 1000 simulations of the variables in the waste incinerator example given no evidence, Lauritzen's evidence (1992), and Olesen's evidence (1991). The results of these simulations are given in Tables 3.10, 3.11, and 3.12

and Figures 3.9, 3.10, and 3.11. Each table presents the probabilities, means and variances of the marginal distributions of the variables for which we have no evidence. The true values are given in brackets. Each figure graphs the simulated marginal distribution of each continuous variable for which we have no evidence. We provide histograms and estimated probability density functions for each variable (with the exception of $E$ which is too spiky for this to make sense). The histograms were generated using the built-in *chart* procedure in *SAS* and were plotted in *Mathematica*. An appropriate routine could equally have been written from scratch in *Mathematica*. The estimated probability density functions were generated using custom-built kernel smoothing techniques in *SAS*. Again, *Mathematica* could equally have been used. A Gaussian kernel was employed and the band-width for each distribution was chosen to display the distribution in its best light. This subjective choice of band-width was chosen with reference to the true marginal probability density functions described in Chapter 4. It was decided to use a subjective band-width in order to put forward the best case for the simulated data since it is the simulation technique which is "on trial" rather than the smoothing technique.

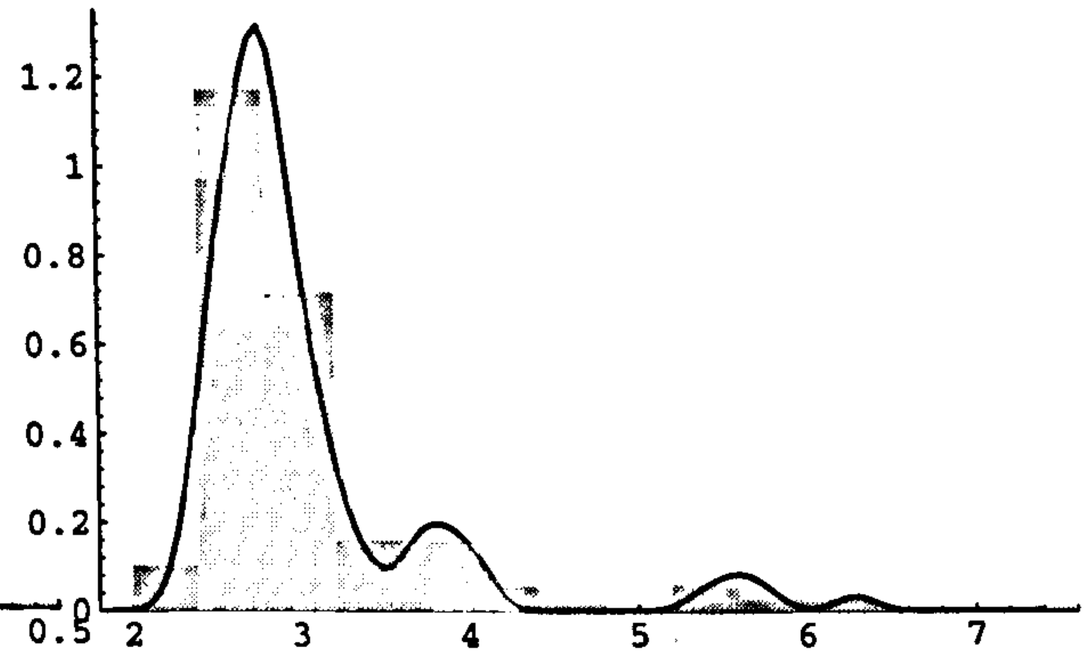
| Variable | Mean | Variance |
|---|---|---|
| C: $CO_2$ in Emission | −1.8645 (−1.8500) | 0.2334 (0.2575) |
| D: Emission of Dust | 3.0350 (3.0393) | 0.5923 (0.5929) |
| E: Filter Efficiency | −3.2501 (−3.2536) | 0.5168 (0.5025) |
| L: Light Penetrability | 1.5076 (1.4804) | 0.3926 (0.3982) |
| Mi: Metal in Waste | −0.2161 (−0.2143) | 0.2094 (0.2105) |
| M0: Emission of Metal | 2.8174 (2.8250) | 0.7208 (0.7401) |

| Variable ($I$) | $i$ | $P(I = i)$ | $\bar{i}$ | $P(I = \bar{i})$ |
|---|---|---|---|---|
| B: Burning Regime | Stable | 0.8490 (0.8500) | Unstable | 0.1510 (0.1500) |
| F: Filter State | Intact | 0.9480 (0.9500) | Defective | 0.0520 (0.0500) |
| W: Type of Waste | Industrial | 0.2850 (0.2857) | Household | 0.7150 (0.7143) |

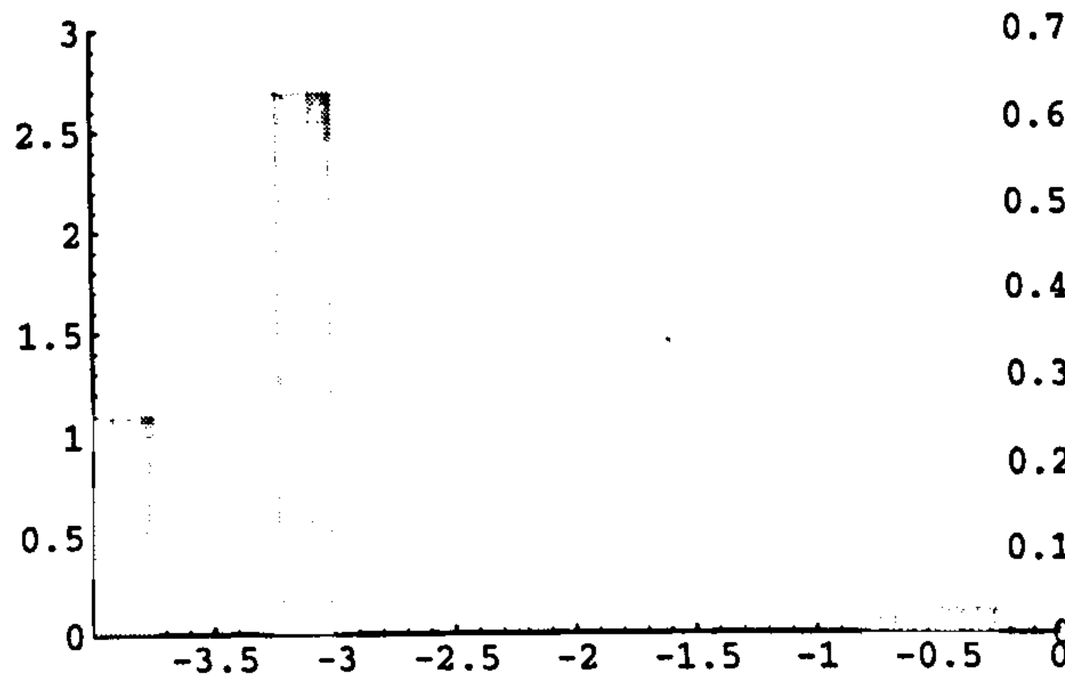
Table 3.10: The probabilities, means and variances of the simulated variables in the waste incinerator example.



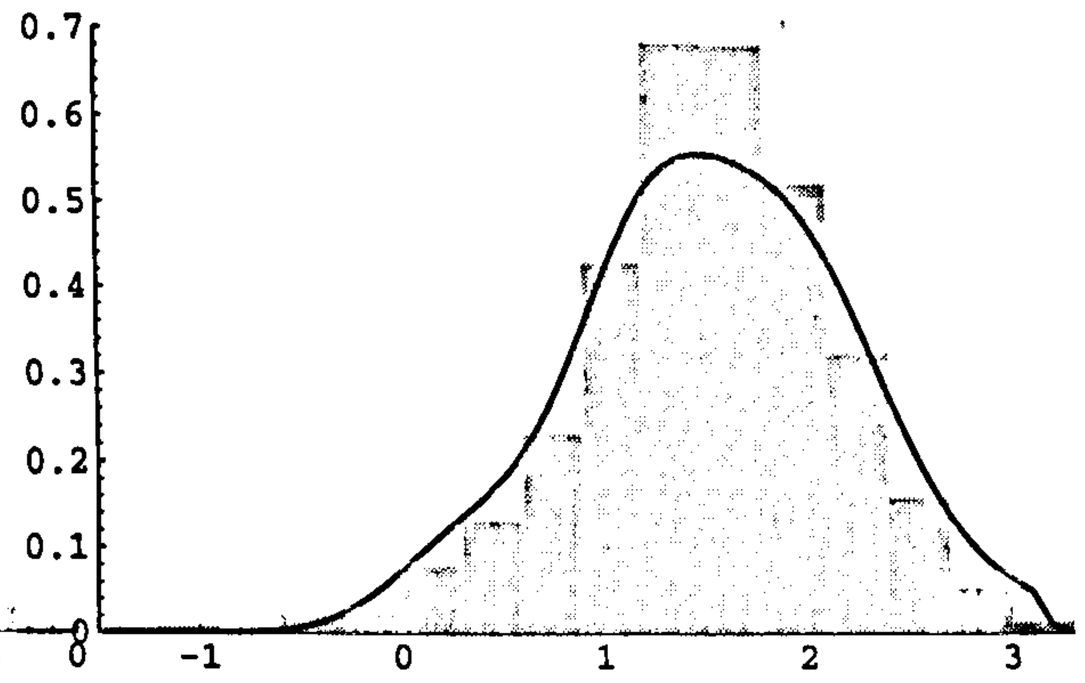

$C$ : $CO_2$ in emission



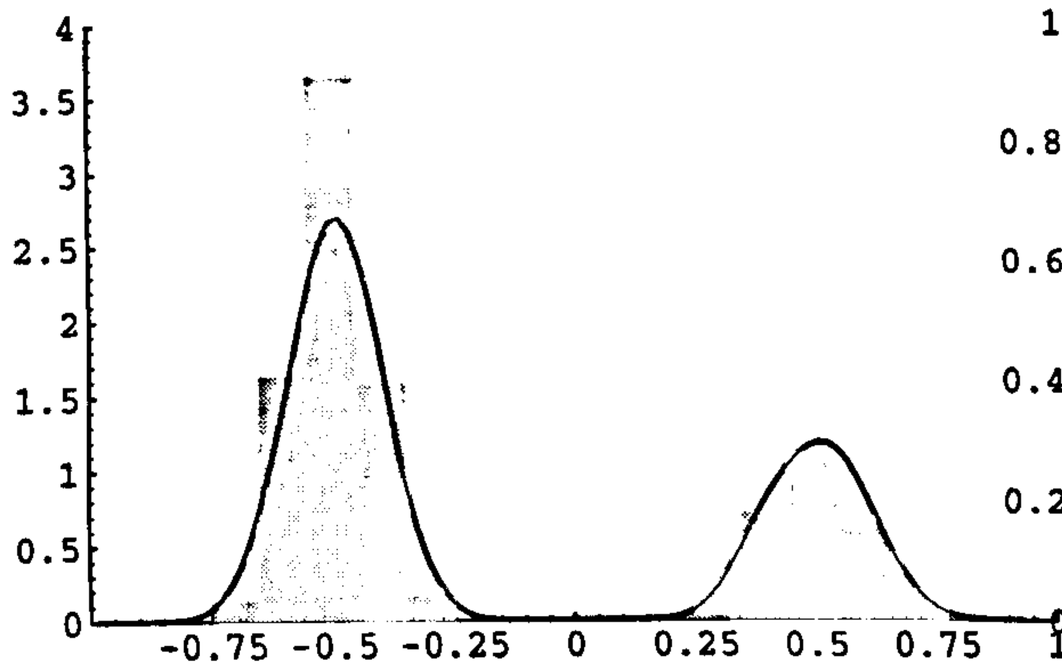

$D$ : Dust in emission



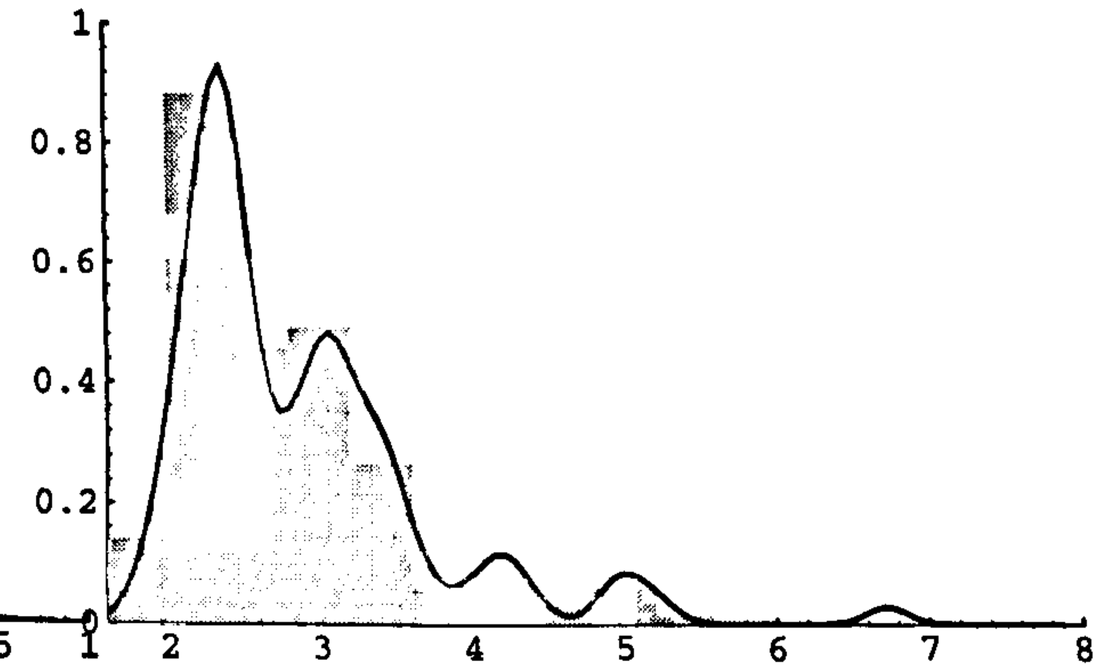

$E$ : Filter efficiency




$L$ : Light penetrability

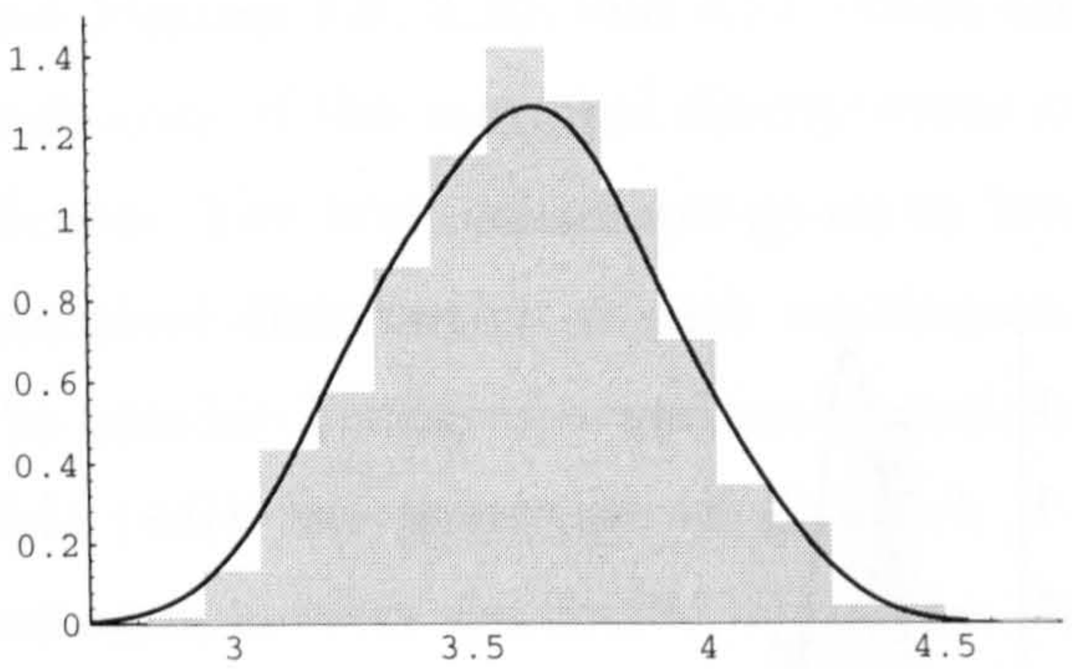


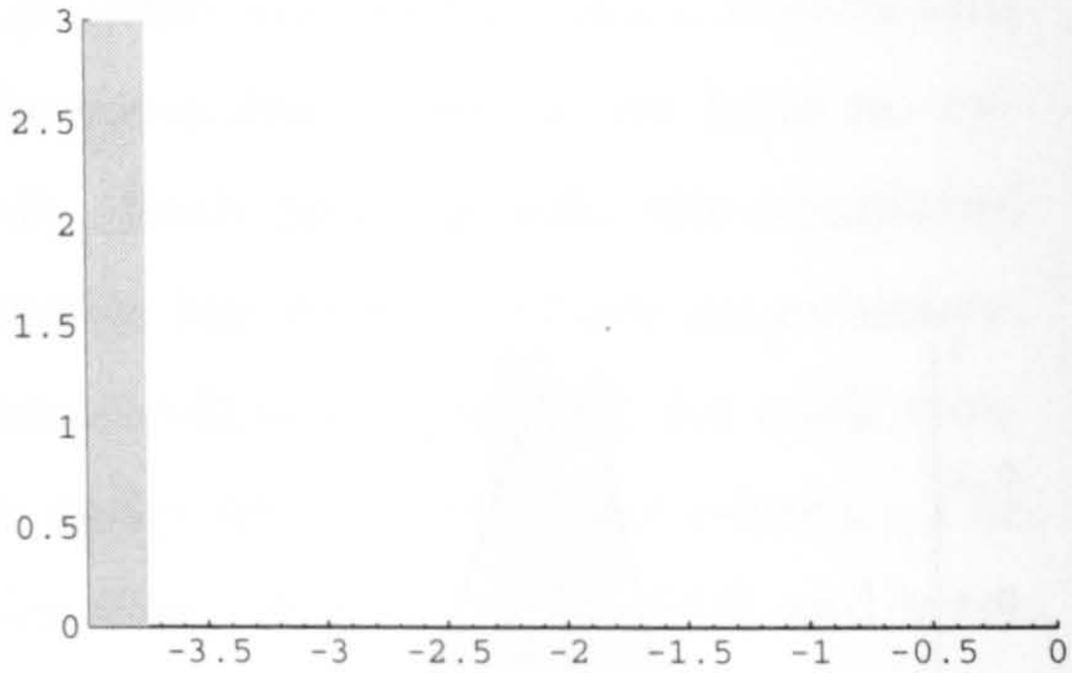

$Mi$ : Metal in waste



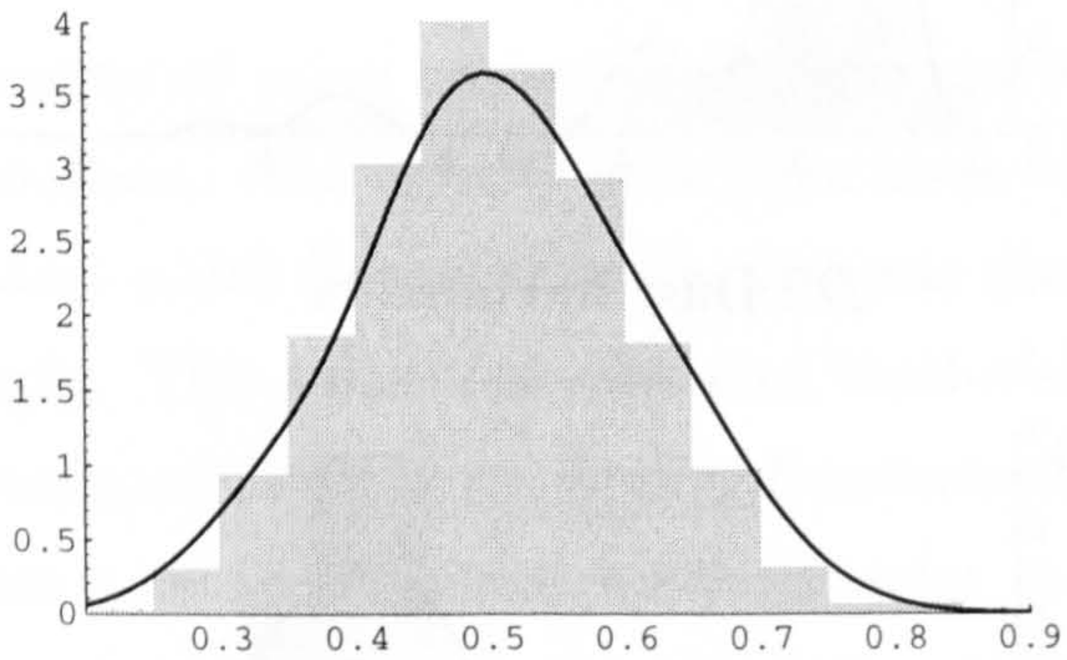

$M0$ : Emission of metal

Figure 3.9: Graphs of the marginal distributions of the simulated continuous variables in the waste incinerator problem.

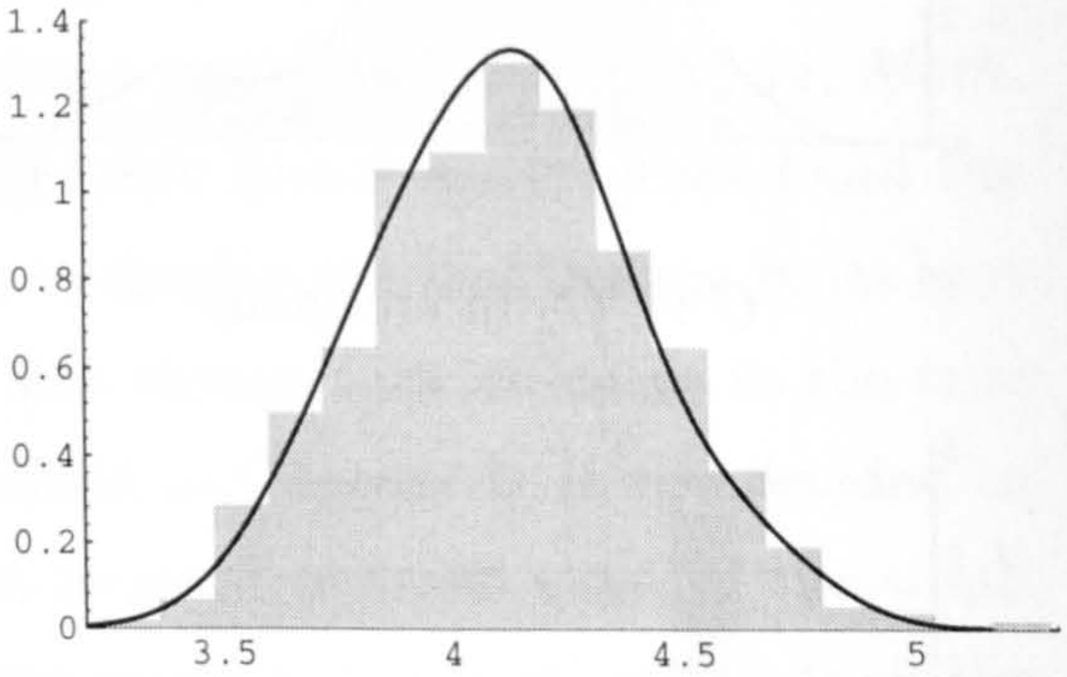



$D$ : Dust in emission




$E$ : Filter efficiency




$Mi$ : Metal in waste

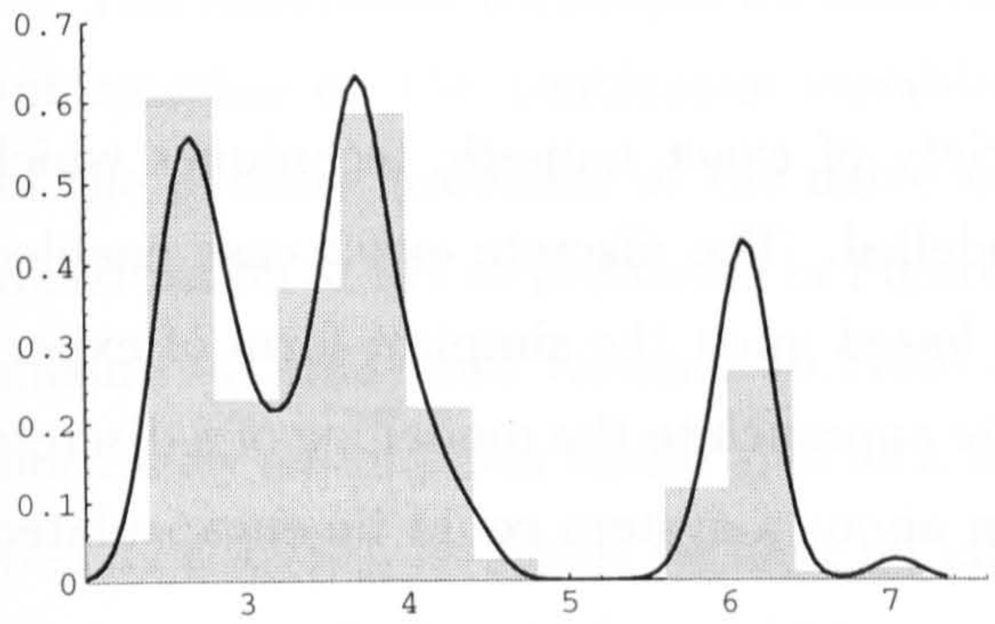
$M0$ : Emission of metal

Figure 3.10: Graphs of the marginal distributions of the simulated continuous variables in the waste incinerator problem given Lauritzen's evidence.

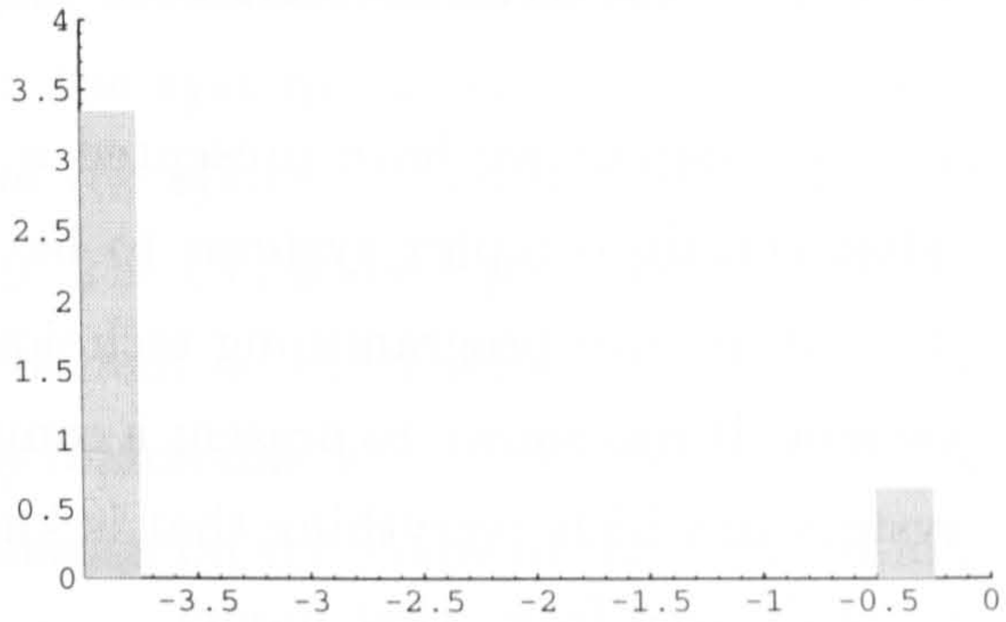
| **Variable** | **Mean** | **Variance** |
|---|---|---|
| D: Emission of Dust | 3.6235 (3.6077) | 0.0820 (0.1062) |
| E: Filter Efficiency | −3.8997 (−3.8983) | 0.0001 (0.0058) |
| Mi: Metal in Waste | −0.5008 (0.5000) | 0.0098 (0.0100) |
| M0: Emission of Metal | 4.1248 (4.1077) | 0.0968 (0.1182) |

| **Variable** $(I)$ | $i$ | $P(I=i)$ | $\bar{i}$ | $P(I=\bar{i})$ |
|---|---|---|---|---|
| B: Burning Regime | Stable | 0.0000 (0.0123) | Unstable | 1.0000 (0.9877) |
| F: Filter State | Intact | 1.0000 (0.9995) | Defective | 0.0000 (0.0005) |

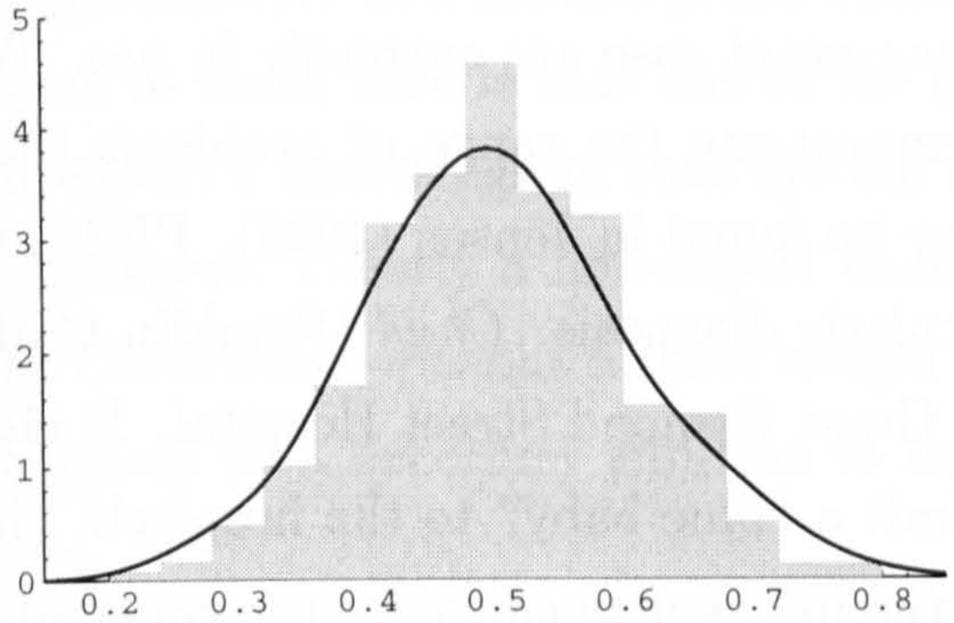
Table 3.11: The probabilities, means and variances of the simulated variables in the waste incinerator example given Lauritzen's evidence.



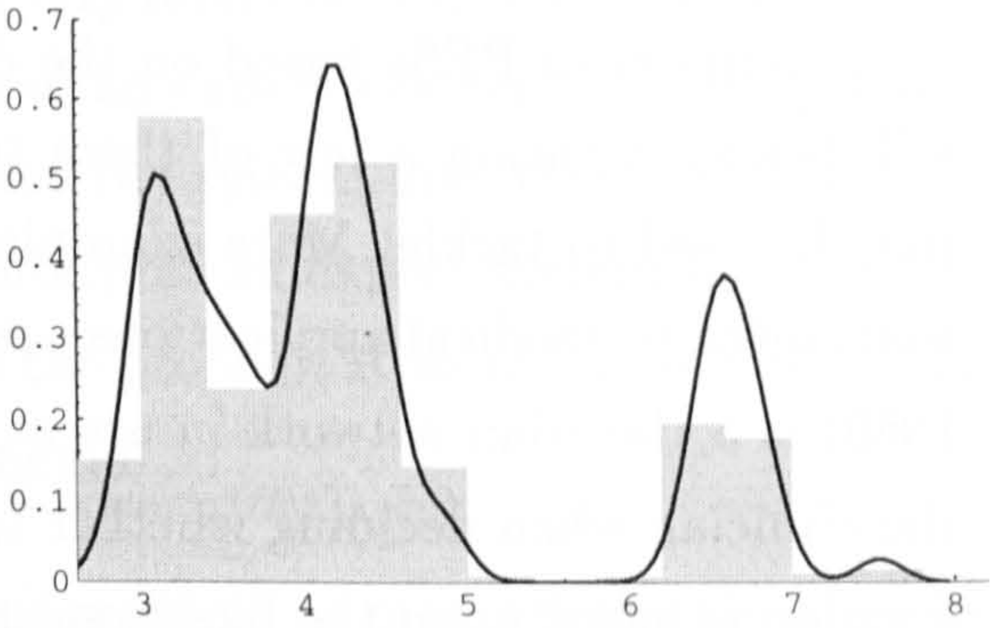
$D$ : Dust in emission



$E$ : Filter efficiency



$Mi$ : Metal in waste



$M0$ : Emission of metal

Figure 3.11: Graphs of the marginal distributions of the simulated continuous variables in the waste incinerator problem given Olesen's evidence.

| **Variable** | **Mean** | **Variance** |
|---|---|---|
| D: Emission of Dust | 3.7623 (3.7745) | 1.3830 (1.7362) |
| E: Filter Efficiency | −3.3288 (−3.1504) | 1.6739 (2.0616) |
| Mi: Metal in Waste | −0.5028 (0.5000) | 0.0091 (0.0100) |
| M0: Emission of Metal | 4.2668 (4.2745) | 1.4014 (1.7482) |

| **Variable** $(I)$ | $i$ | $P(I = i)$ | $\bar{i}$ | $P(I = \bar{i})$ |
|---|---|---|---|---|
| B: Burning Regime | Stable | 0.5070 (0.6424) | Unstable | 0.4930 (0.3576) |
| F: Filter State | Intact | 0.8370 (0.7858) | Defective | 0.1630 (0.2142) |

Table 3.12: The probabilities, means and variances of the simulated variables in the waste incinerator example given Olesen's evidence.

## 3.16 Conclusions

In this chapter we have presented a variety of exact numeric techniques which allow certain complex systems to be modelled. The discrete exact case enabled us to introduce programming techniques based upon the simplest form of expert system. It was shown to present a complete approach to the modelling of a discrete system in which everything that is known about a system could be encapsulated in, or derived from, that model.

A number of PESs based on the discrete exact case are currently in use. We will briefly mention a few of them to demonstrate the range of problems they may be used to tackle. More examples may be found in Jensen (1996). PESs are well suited to medical applications - particularly diagnosis. *Child* (Franklin *et al.*, 1989) is a Bayesian network in use at the Great Ormond Street Hospital. It aids the clinician when deciding whether to admit a "blue baby" to the hospital. The decision is based upon the likelihood that the infant may suffer from a congenital heart disease. Like the fictitious chest clinic example, *MUNIN* (Andreassen *et al.*, 1989) is a system used to diagnose disease - in this case diseases affecting the muscles and nerves. The *Pathfinder* system assists community pathologists with the diagnosis of lymph-node pathology (Heckerman *et al.*, 1992). It has been integrated into the commercial system *Intellipath*.

At the Danish Blood Type Laboratory the *BOBLO* system (Rasmussen, 1995) is used to help verify the parentage of Jersey cattle through blood type identification. The *VISTA* system (Horvitz & Barry, 1995) is used by NASA to filter and display information on the propulsion system of the space shuttle. The *PRESS* system (Aitken & Gammerman, 1989; Aitken *et al.*, 1996) has been applied in forensic science for the statistical profiling of offenders. Probably the most widely distributed PES is contained within Microsoft's Windows 95 operating system. It is a system which trouble-shoots printing problems (Heckerman *et al.*, 1995).

In contrast to the discrete exact case, the modelling of the mixed case using CG-potentials allows us to enter the complete set of information we have about a given system into the model but only allows a subset of the information on the system to be derived from that model. In particular while, as in the discrete case, we may obtain the entire set of information we require on the discrete variables, the information on the continuous variables is limited to their means and variances.

The simulation technique we describe goes towards our goal of obtaining more information on the continuous variables in the system. It also highlights some of the possible problems of the naive use of the system. Consider the simulated distribution of $Mi$ as presented in Figure 3.9. The CG-scheme provides us with the mean $(-0.2143)$ and variance $(0.2105)$ of $Mi$ and no other information. A naive normality assumption would give us a fairly reasonable 95% confidence interval of $(-1.114, 0.685)$, however, we have no information on the shape of the distribution. The mean, which naively would be our best guess at a typical realisation of the distribution, in fact lies at a point which will very rarely be realised. The important point to make here is that one of the goals of an expert system is to encapsulate an expert's knowledge in that system so that the system may then be used by a non-expert. Clearly a non-expert cannot interpret an expert system based upon the CG-scheme without further insight into the problem at hand. Simulations will therefore be a necessary addition to the scheme.

The complexity of some of the operations required to program the CG-scheme may result in loss of accuracy. For example the need to apply the logarithmic and exponential operators when converting between canonical and moment characteristics and back again. *Mathematica*'s high level of accuracy appears to cope with these problems although $C$ has been found to be less kind.

Nevertheless the CG-scheme accomplishes what it sets out to do, namely to model probabilities, means and variances, in a most elegant and revolutionary way. It represents the first major approach to the modelling of the mixed case and, taken with its limitations, performs very well. It may be of most use where speed is of the essence since it should be faster than any symbolic or approximate technique which seeks to offer a fuller set of information.

In the remaining chapters we will seek to build upon the modelling capabilities of probabilistic expert systems not only in terms of the class of models they may tackle, but also in terms of the amount of information we may derive from them. *Mathematica*'s usefulness in the tackling of the discrete exact case and the CG-scheme show that it is a very flexible environment for the development of probabilistic expert systems. In later chapters we will make more use of the features with which *Mathematica* is most commonly associated. In particular its symbolic and graphical capabilities.

# Chapter 4

# Symbolic Techniques

## 4.1 Introduction

In the last section we investigated two classes of PES and described exact numeric techniques which facilitated their modelling. The first class of PES, the *discrete exact case*, was built under the assumption that the probabilities on every discrete variable could be represented by known real numbers. We were therefore able to program the system in a wide range of different languages using only computer arithmetic to perform the required calculations. The second class of PES comprised strongly decomposable graphs and *Conditional Gaussian distributions.* These models were built under the assumption that the mean and variance of the conditional distribution of each continuous variable given its parents and the probabilities on every discrete variable could be represented by known real numbers. Under this situation we were, again, able to program the system using only computer arithmetic to perform the required calculations.

In this section we attempt to go one step further by relaxing the condition that the probabilities, means and variances of distributions need be represented numerically. Instead we employ the use of computer algebra to allow the symbolic manipulation of the potentials. We investigate two discrete examples and one mixed example. In its simplest form we show how we may model the discrete exact case in *Mathematica* where a probability is represented by a symbolic parameter rather than a number. We later examine the situation where a discrete variable takes a probability distribution which may not be represented by a simple table of numbers.

In our mixed example we revisit Lauritzen's waste incinerator example. Lau-

ritzen's approach of modelling the means and variances of the continuous variables in the mixed graphical association model rather than their probability density functions may in hindsight seem slightly strange. It should be noted however that when armed with computer arithmetic alone it is unclear how exact methodologies may best be constructed to deal with probability densities. The CG-scheme fits in well with the discrete exact case. Both use tables of numbers as data structures for their potentials. In the discrete exact case potential tables are used. In the CG-scheme canonical and moment characteristics are stored as tables of numbers. We will now take the more direct approach in this chapter and model the variables using their probability density functions. This maximises the amount of information we will be able to derive from the model. We show how evidence of both a numeric and a symbolic type may be entered into our system.

## 4.2 Genetic Counselling Example

In this section we introduce Spiegelhalter's genetic counselling problem (Spiegelhalter, 1990). This problem was originally designed to show how the techniques used in probabilistic expert systems may be applied to problems considered in pedigree analysis. It does, however, also provide a relatively simple environment in which we may investigate how we may extend the discrete exact case to cover situations where a symbolic parameter is to be employed. In the problem we assume that we are interested in measuring the probability of the occurrence of some genetic trait in the offspring of a set of parents. In order to keep the problem as general as possible it is not stated what that genetic trait might be, nor what species the parents belong to. The only information we have is part of the family tree (genealogy or pedigree) of the parents and the observed traits of individuals within that tree.

In order to explain the problem fully we must first define some basic terminology and probability assignments relevant to the field of genetics. Consider some genetic trait of an individual. The set of possible states which that trait may possess is known as the trait's *phenotype*. The phenotype of a trait is governed by the individual's genetic makeup. For the purposes of this example we will assume that the trait in question is governed by a single pair of genes and that each gene may be one of two possible types known as *alleles*. We will denote the alleles $s$ and $S$. One gene is a copy of a randomly chosen member of the father's relevant pair of genes, and one is derived from the mother. The *genotype* of the trait refers

to the unordered pair of alleles which comprise the pair of genes. It may therefore be $ss$, $sS$, or $SS$. If an individual with genotype $sS$ or $SS$ is affected by the trait then $S$ is the allele for the trait, and is said to be *dominant.* If only individuals with the genotype $SS$ are affected by the trait then $S$ is said to be *recessive* and individuals with the genotype $sS$ are said to be *carriers* of the trait since they can pass it on to their offspring but are not affected themselves. If the pair of genes has two alleles the same, either $ss$ or $SS$ it is said to be *homozygous*. If the pair of genes is $sS$ then it is said to be *heterozygous*. Let us suppose, in our example, that we are unable to determine the trait's genotype but we are able to determine its phenotype.

In our example we assume that the trait is recessive. We may therefore determine the probability of possessing a particular phenotype given its genotype to be as follows:

$$\begin{aligned} P(\text{affected} \mid ss) &= 0 \\ P(\text{affected} \mid sS) &= 0 \\ P(\text{affected} \mid SS) &= 1 \end{aligned}$$

In order to assign the transmission of a genotype from parents to child we will use the model of *Mendelian segregation.* This states that each parent independently contributes a random allele chosen with probability $\frac{1}{2}$. In order to determine the genotype of the *founders*, those individuals whose parents are not contained in the genealogy, we will assume that their genotypes are independent and randomly allocated according to their population frequency. Let $P(s) = p$ and $P(S) = 1 - p = q$, then *Hardy-Weinberg equilibrium* states that the genotype distributions are:

$$P(ss) = p^2 \quad P(sS) = 2pq \quad P(SS) = q^2$$

This assumption introduces an unknown parameter $p$ into the problem and hence dictates the application of a symbolic method.

Figure 4.1 represents the causal probabilistic network for the genetics counselling problem. We have a set of individuals { Arthur, Betty, Charles, Derek, Edith, Florence, George, Hilda, Ivan, John } with respective genotypes $A$-$J$ and phenotypes $A'$-$J'$ for a particular trait. The nodes of the CPN denote these genotypes and phenotypes. We are interested in the genotype $I$ and phenotype $I'$ of

the trait for Ivan, the prospective offspring of Charles and Florence. It is known that Florence's nephew John has inherited the trait. The genotypes, $A$-$J$, may take three levels - $ss$, $sS$, and $SS$. The phenotypes, $A'$-$J'$, may take one of two levels- normal ($n$) or affected ($\bar{n}$). Probability tables for the genotypes of the founders $\{A, B, E, H\}$ and conditional probability tables for the genotypes of the offspring $\{C, D, F, G, I, J\}$ and phenotypes $A'$-$J'$ are given in Table 4.1.

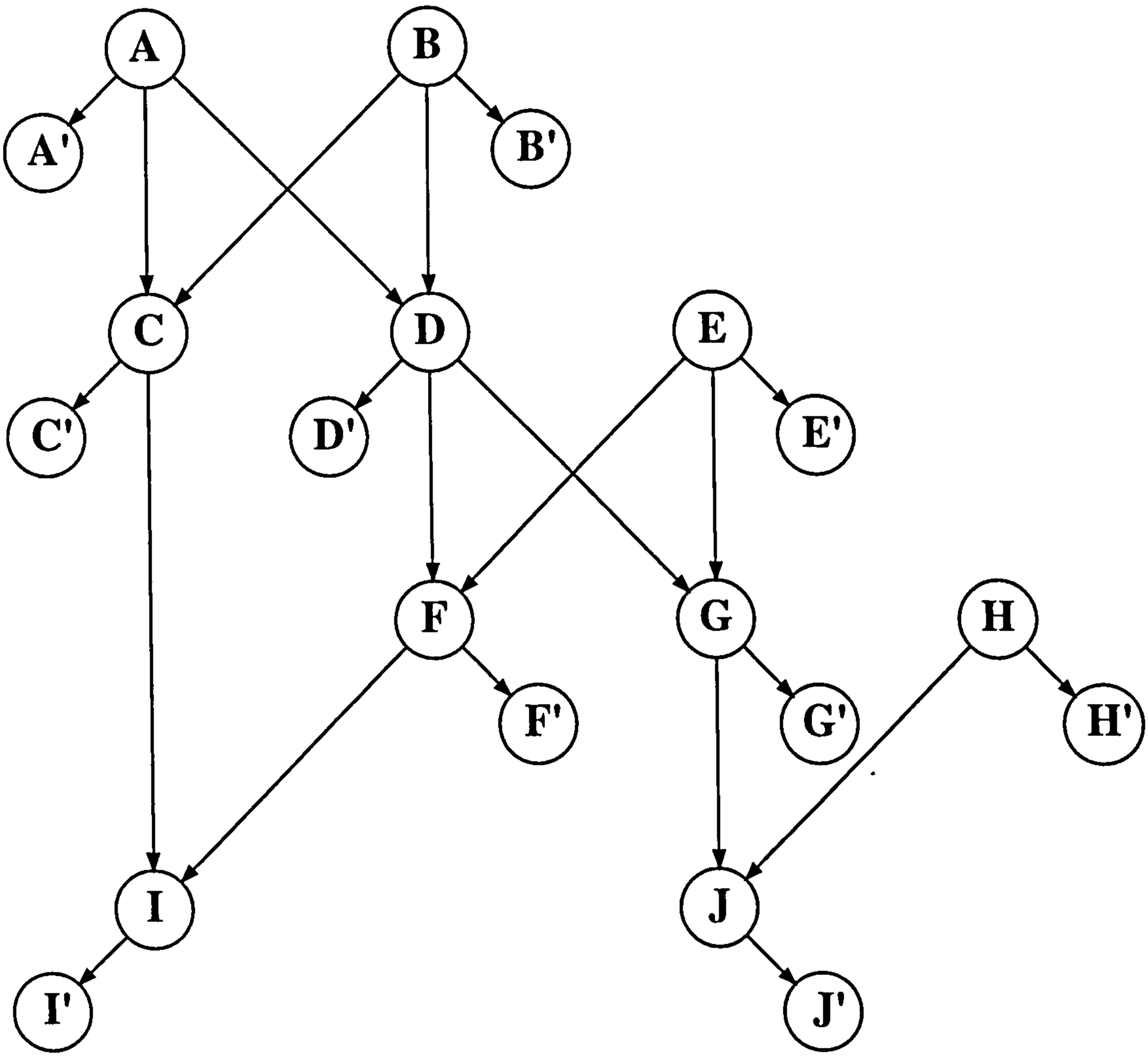



Figure 4.1: The CPN illustrating the genetics counselling problem.

Figure 4.2 displays the junction tree formed from the CPN of Figure 4.1. If we know that John is the only affected individual in the pedigree, given that Ivan has not yet been conceived, then we have information on the phenotypes $A'$-$H'$ and $J'$. We may therefore simplify the junction tree to that of Figure 4.3. To take account of this evidence we may update the conditional probability tables. For those individuals unaffected by the disease the homozygous recessive genotype $SS$ becomes impossible. We may therefore zero this out in the corresponding

**Founder Genotypes:**

$P(X = x)$ **for** $X \in \{A, B, E, H\}$

| | $x$ | |
|---|---|---|
| $ss$ | $sS$ | $SS$ |
| $p^2$ | $2pq$ | $q^2$ |

**Offspring Genotypes:**

$P(X = x \mid Pa(X))$ **for** $X \in \{C, D, F, G, I, J\}$, $Pa(X) = \{P_1, P_2\}$

| | | $x$ | | |
|---|---|---|---|---|
| $P_1$ | $P_2$ | $ss$ | $sS$ | $SS$ |
| $ss$ | $ss$ | 1 | 0 | 0 |
| $ss$ | $sS$ | $\frac{1}{2}$ | $\frac{1}{2}$ | 0 |
| $ss$ | $SS$ | 0 | 1 | 0 |
| $sS$ | $ss$ | $\frac{1}{2}$ | $\frac{1}{2}$ | 0 |
| $sS$ | $sS$ | $\frac{1}{4}$ | $\frac{1}{2}$ | $\frac{1}{4}$ |
| $sS$ | $SS$ | 0 | $\frac{1}{2}$ | $\frac{1}{2}$ |
| $SS$ | $ss$ | 0 | 1 | 0 |
| $SS$ | $sS$ | 0 | $\frac{1}{2}$ | $\frac{1}{2}$ |
| $SS$ | $SS$ | 0 | 0 | 1 |

**Phenotypes:**

$P(X = x \mid Pa(X))$ **for** $X \in \{A', B', C', D', E', F', G', H', I', J'\}$, $Pa(X) = \{P\}$

| | $x$ | |
|---|---|---|
| $P$ | $n$ | $\bar{n}$ |
| $ss$ | 1 | 0 |
| $sS$ | 1 | 0 |
| $SS$ | 0 | 1 |

Table 4.1: The initial conditional probability tables of the genotypes $A$-$H$ and phenotypes $A'$-$H'$ in the genetic counselling example.

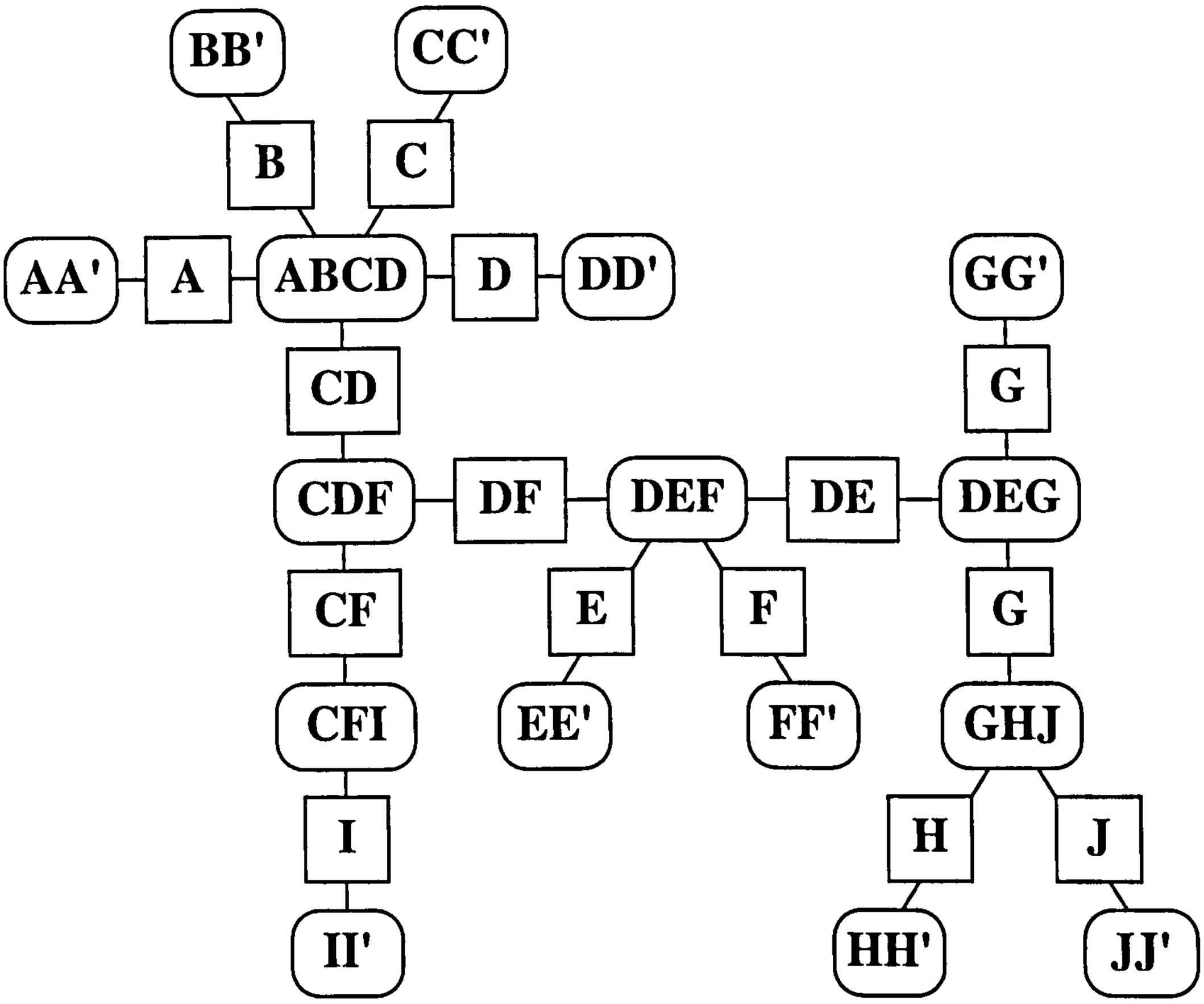



Figure 4.2: The junction tree illustrating the genetics counselling problem.

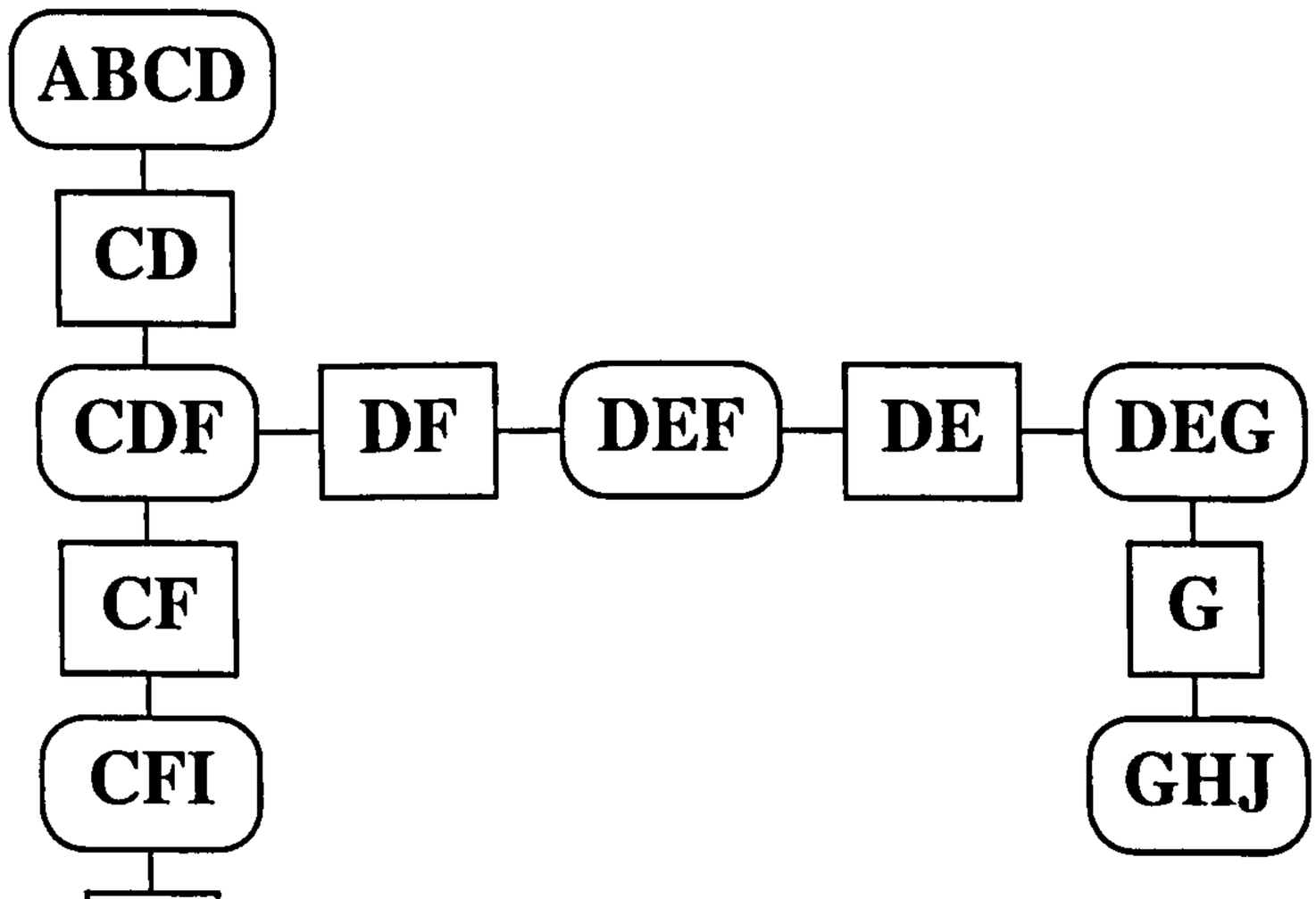



Figure 4.3: The junction tree illustrating the genetics counselling problem given evidence on the phenotypes $A'$-$H'$.

probability tables. Since we know John has the disease we know that his genotype is $SS$. We may therefore zero out the probability that $J$ is $ss$ or $sS$ in the conditional probability table for $J \mid G, H$. These simplifications are not strictly necessary. We could use the junction tree of Figure 4.2 and enter the phenotypic evidence into the system using our usual techniques. Simplifying the junction tree will make the example easier to describe and reduces the number of computations required.

While the programming of the genetic counselling example in a numeric language such as *Dyalog APL* or *C* would be a major undertaking, its programming in a symbolic language like *Mathematica* is relatively simple. The basic mathematical functions in *Mathematica* have been designed to take both numeric and symbolic expressions. Therefore the functions we described in Chapter 3 which enabled the programming of a discrete exact PES may also be applied to certain symbolic PESs. In theory, any discrete PES with finite state space and real or symbolic probabilities may be programmed using the *Mathematica* functions described in Chapter 3. In practice, however, there will be both memory and time constraints on the implementation of some PESs. The functions should probably also be changed to incorporate a simplification phase every time a symbolic calculation is performed. This will minimise the complexity of the symbolic expressions generated but will have a trade-off in terms of speed. For genetics examples the *Mathematica* command `Factor` may be used as an appropriate simplification function. `Factor` factors a polynomial with exact integer or rational coefficients. The exponents of the variables may be positive integers or linear combinations of symbolic expressions.

The initial probability distributions of each random variable in the genetic counselling example as defined in Table 4.1 may, given evidence on $A'$-$H'$ and $J'$, be defined as potentials as follows:

```
inita = {{a}, {p^2, 2p(1 − p), 0}}
initb = {{b}, {p^2, 2p(1 − p), 0}}
initc = {{c, a, b}, {{{1, 1/2, 0}, {1/2, 1/4, 0}, {0, 0, 0}},
        {{0, 1/2, 0}, {1/2, 1/2, 0}, {0, 0, 0}}, {{0, 0, 0}, {0, 0, 0}, {0, 0, 0}}}}
initd = {{d, a, b}, {{{1, 1/2, 0}, {1/2, 1/4, 0}, {0, 0, 0}},
        {{0, 1/2, 0}, {1/2, 1/2, 0}, {0, 0, 0}}, {{0, 0, 0}, {0, 0, 0}, {0, 0, 0}}}}
```

$$
\begin{aligned}
\texttt{inite} &= \{\{\texttt{e}\},\{\texttt{p\^{}2},2\texttt{p}(1-\texttt{p}),0\}\} \\
\texttt{initf} &= \{\{\texttt{f},\texttt{d},\texttt{e}\},\{\{\{1,1/2,0\},\{1/2,1/4,0\},\{0,0,0\}\}, \\
&\quad \{\{0,1/2,0\},\{1/2,1/2,0\},\{0,0,0\}\},\{\{0,0,0\},\{0,0,0\},\{0,0,0\}\}\}\} \\
\texttt{initg} &= \{\{\texttt{g},\texttt{d},\texttt{e}\},\{\{\{1,1/2,0\},\{1/2,1/4,0\},\{0,0,0\}\}, \\
&\quad \{\{0,1/2,0\},\{1/2,1/2,0\},\{0,0,0\}\},\{\{0,0,0\},\{0,0,0\},\{0,0,0\}\}\}\} \\
\texttt{inith} &= \{\{\texttt{h}\},\{\texttt{p\^{}2},2\texttt{p}(1-\texttt{p}),0\}\} \\
\texttt{initi} &= \{\{\texttt{i},\texttt{c},\texttt{f}\},\{\{\{1,1/2,0\},\{1/2,1/4,0\},\{0,0,0\}\}, \\
&\quad \{\{0,1/2,0\},\{1/2,1/2,0\},\{1,1/2,0\}\}, \\
&\quad \{\{0,0,0\},\{0,1/4,0\},\{0,1/2,0\}\}\} \\
\texttt{initj} &= \{\{\texttt{j},\texttt{g},\texttt{h}\},\{\{\{0,0,0\},\{0,0,0\},\{0,0,0\}\}, \\
&\quad \{\{0,0,0\},\{0,0,0\},\{0,0,0\}\},\{\{0,0,0\},\{0,1/4,0\},\{0,0,0\}\}\}\} \\
\texttt{initi}' &= \{\{\texttt{i}',\texttt{i}\},\{\{1,1,0\},\{0,0,1\}\}\}
\end{aligned}
$$

We may then define a list, `init`, of all the initial probability distributions where:

$$
\begin{aligned}
\texttt{init} &= \{\texttt{inita},\texttt{initb},\texttt{initc},\texttt{initd},\texttt{inite},\texttt{initf},\texttt{initg},\texttt{inith},\texttt{initi}, \\
&\quad \texttt{initj},\texttt{initi}'\}
\end{aligned}
$$

The lists of the random variables in the problem, `globvars`, and the number of levels of these variables, `globnums`, are:

$$
\begin{aligned}
\texttt{globvars} &= \{\texttt{a},\texttt{b},\texttt{c},\texttt{d},\texttt{e},\texttt{f},\texttt{g},\texttt{h},\texttt{i},\texttt{j},\texttt{i}'\} \\
\texttt{globnums} &= \{3,3,3,3,3,3,3,3,3,3,2\}
\end{aligned}
$$

The lists of cliques, separators, and the assignment of each variable to each clique are `cliques`, `seps`, and `assign` respectively where:

$$
\begin{aligned}
\texttt{cliques} &= \{\{\texttt{a},\texttt{b},\texttt{c},\texttt{d}\},\{\texttt{c},\texttt{d},\texttt{f}\},\{\texttt{d},\texttt{e},\texttt{f}\},\{\texttt{d},\texttt{e},\texttt{g}\},\{\texttt{c},\texttt{f},\texttt{i}\},\{\texttt{g},\texttt{h},\texttt{j}\},\{\texttt{i},\texttt{i}'\}\} \\
\texttt{seps} &= \{\{\{\texttt{c},\texttt{d}\},\{1\}\},\{\{\texttt{d},\texttt{f}\},\{2\}\},\{\{\texttt{d},\texttt{e}\},\{3\}\},\{\{\texttt{c},\texttt{f}\},\{4\}\}, \\
&\quad \{\{\texttt{g}\},\{5\}\},\{\{\texttt{i}\},\{6\}\}\} \\
\texttt{assign} &= \{\{\texttt{a},\texttt{b},\texttt{c},\texttt{d}\},\{\},\{\texttt{e},\texttt{f}\},\{\texttt{g}\},\{\texttt{i}\},\{\texttt{h},\texttt{j}\},\{\texttt{i}'\}\}
\end{aligned}
$$

The first half of a palindromic propagation schedule for the problem is `sched` which may be defined as:

$$\texttt{sched} = \{\{\{\texttt{g},\texttt{h},\texttt{j}\},\{\{\texttt{g}\},\{\texttt{5}\}\},\{\texttt{d},\texttt{e},\texttt{g}\}\},\{\{\texttt{d},\texttt{e},\texttt{g}\},\{\{\texttt{d},\texttt{e}\},\{\texttt{3}\}\},\{\texttt{d},\texttt{e},\texttt{f}\}\},$$
$$\{\{\texttt{d},\texttt{e},\texttt{f}\},\{\{\texttt{d},\texttt{f}\},\{\texttt{2}\}\},\{\texttt{c},\texttt{d},\texttt{f}\}\},\{\{\texttt{a},\texttt{b},\texttt{c},\texttt{d}\},\{\{\texttt{c},\texttt{d}\},\{\texttt{1}\}\},\{\texttt{c},\texttt{d},\texttt{f}\}\},$$
$$\{\{\texttt{c},\texttt{d},\texttt{f}\},\{\{\texttt{c},\texttt{f}\},\{\texttt{4}\}\},\{\texttt{c},\texttt{f},\texttt{i}\}\},\{\{\texttt{c},\texttt{f},\texttt{i}\},\{\{\texttt{i}\},\{\texttt{6}\}\},\{\texttt{i},\texttt{i}'\}\}\}$$

With these data definitions in place we may initialise the system using the `initialise` function, propagate a schedule using `prop[sched]`, and normalise the resulting clique and separator potentials using `norm` to take account of the evidence on $A'$-$H'$ and $J'$. Alternatively we could initialise the system, propagate the first half of the propagation schedule, determine the normalisation constant from the potential on $(I, I')$ and normalise this potential before propagating the second half of the propagation schedule. The second method is computationally more efficient than the first since less operations are required to normalise the system. Moreover, suppose that we only require the probability that Ivan will be affected. With the second method we need not pass the second half of the propagation schedule - we simply need to marginalise $(I, I')$ with respect to $I$.

Having passed the first half of the propagation schedule the potential on clique $(I, I')$ is found to be as given in Table 4.2:

| $I'$ | $I$ | |
|---|---|---|
| | $n$ | $\bar{n}$ |
| $ss$ | $\frac{(1-p)^2p^4(14+8p-p^2)}{32}$ | 0 |
| $sS$ | $\frac{(1-p)^2p^4(13+p-4p^2)}{32}$ | 0 |
| $SS$ | 0 | $\frac{(1-p)^2p^4(3-p-p^2)}{32}$ |

Table 4.2: The potential table on $(I, I')$ following the collect evidence phase.

Marginalising this potential with respect to both $I$ and $I'$ the normalisation constant may be found to be:

$$\frac{(3-p)(1-p)^2p^4(5+3p)}{16} \tag{4.1}$$

Dividing the potential on $(I, I')$ by the normalisation constant and propagating the distribute evidence phase of the propagation schedule the system is normalised and reaches equilibrium. We may then determine the marginal probabilities of each of the genotypes $A$-$J$ and the phenotype $I'$. These are given in Table 4.3.

**Genotypes:**

$P(X = x)$ **for** $X \in \{A, B, C, D, E, F, G, H, I, J\}$

| $X$ | x $ss$ | $sS$ | $SS$ |
|---|---|---|---|
| $A$ | $\frac{2(p-7)p}{(p-3)(5+3p)}$ | $\frac{p^2+10p-15}{(p-3)(5+3p)}$ | 0 |
| $B$ | $\frac{2(p-7)p}{(p-3)(5+3p)}$ | $\frac{p^2+10p-15}{(p-3)(5+3p)}$ | 0 |
| $C$ | $\frac{(1+p)(5+p)}{(3-p)(5+3p)}$ | $\frac{2(2p^2+p-5)}{(p-3)(5+3p)}$ | 0 |
| $D$ | $\frac{2(1+p)}{5+3p}$ | $\frac{3+p}{5+3p}$ | 0 |
| $E$ | $\frac{2p(3+p)}{(3-p)(5+3p)}$ | $\frac{5p^2+2p-15}{(p-3)(5+3p)}$ | 0 |
| $F$ | $\frac{p^2-3p-6}{(p-3)(5+3p)}$ | $\frac{2p^2-p-9}{(p-3)(5+3p)}$ | 0 |
| $G$ | 0 | 1 | 0 |
| $H$ | 0 | 1 | 0 |
| $I$ | $\frac{p^2-8p-14}{2(p-3)(5+3p)}$ | $\frac{4p^2-p-13}{2(p-3)(5+3p)}$ | $\frac{p^2+p-3}{2(p-3)(5+3p)}$ |
| $J$ | 0 | 0 | 1 |

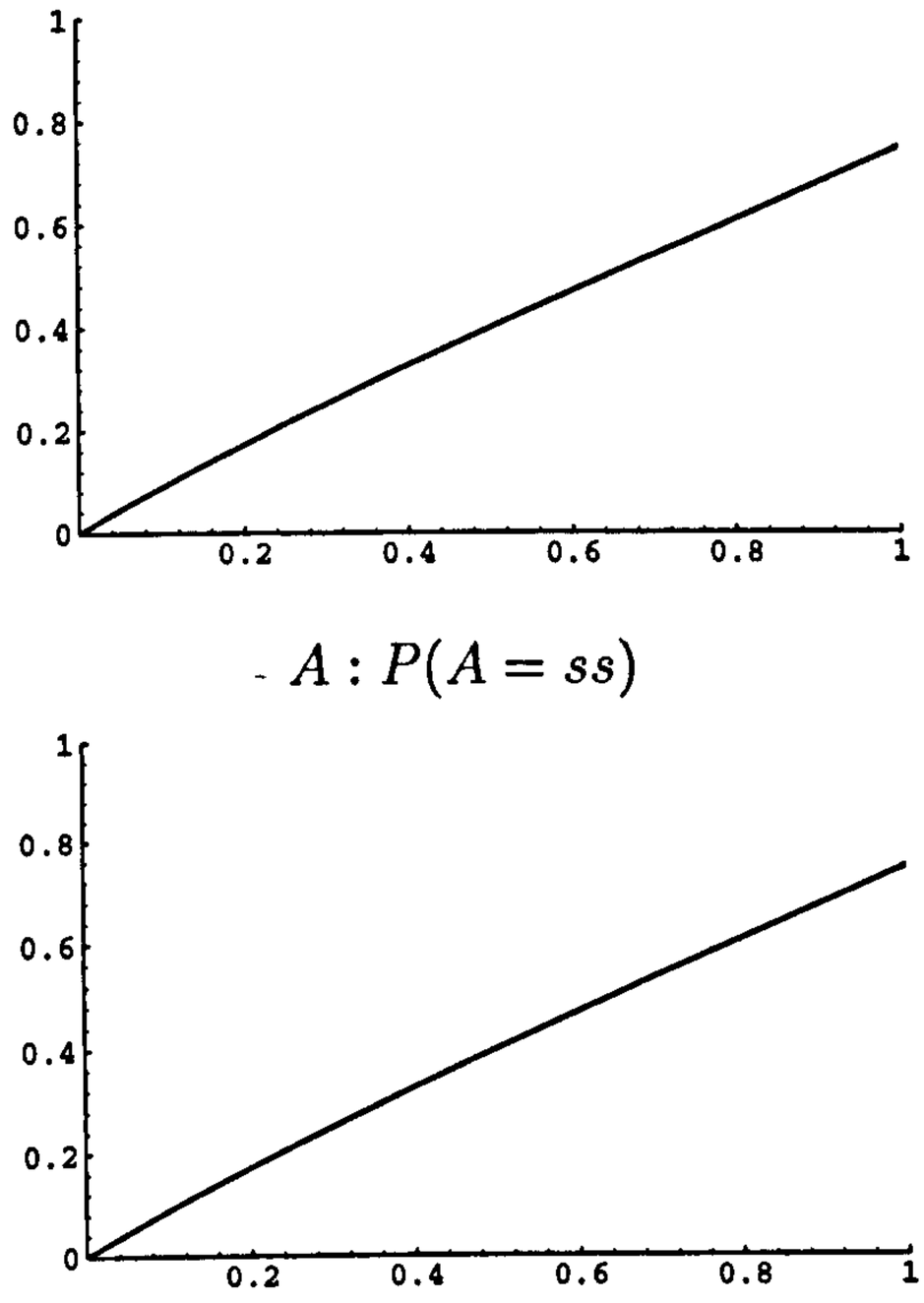
**Phenotype:**

$P(X = x)$ **for** $X \in \{I'\}$

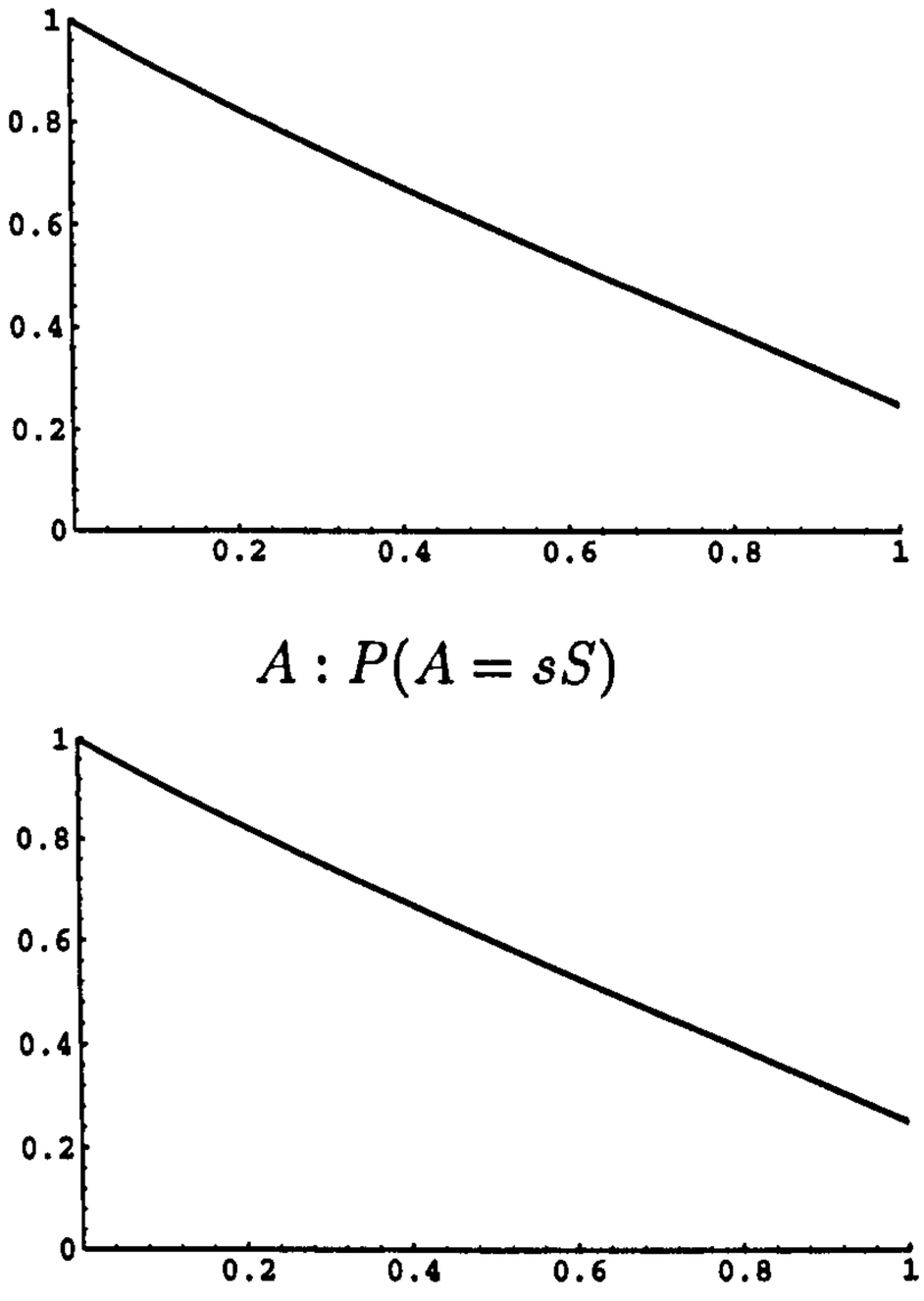
| $X$ | x $n$ | $\bar{n}$ |
|---|---|---|
| $I'$ | $\frac{5p^2-9p-27}{2(p-3)(5+3p)}$ | $\frac{p^2+p-3}{2(p-3)(5+3p)}$ |

Table 4.3: The marginal distributions on the genotypes $A$-$J$ and the phenotype $I'$.

By Table 4.3 only the marginal distributions of the genotypes $G$, $H$ and $J$ are independent of the level of $p$. Of the other variables the genotypes $A$-$F$ may either have levels $ss$ or $sS$, $I$ may take levels $ss$, $sS$, or $SS$ and the phenotype $I'$ may be either $n$ or $\bar{n}$. The marginal probabilities of the possible levels of $A$-$F$, $I$ and $I'$ are plotted in Figures 4.4, 4.5 and 4.6 for varying levels of $p$.




$A : P(A = ss)$



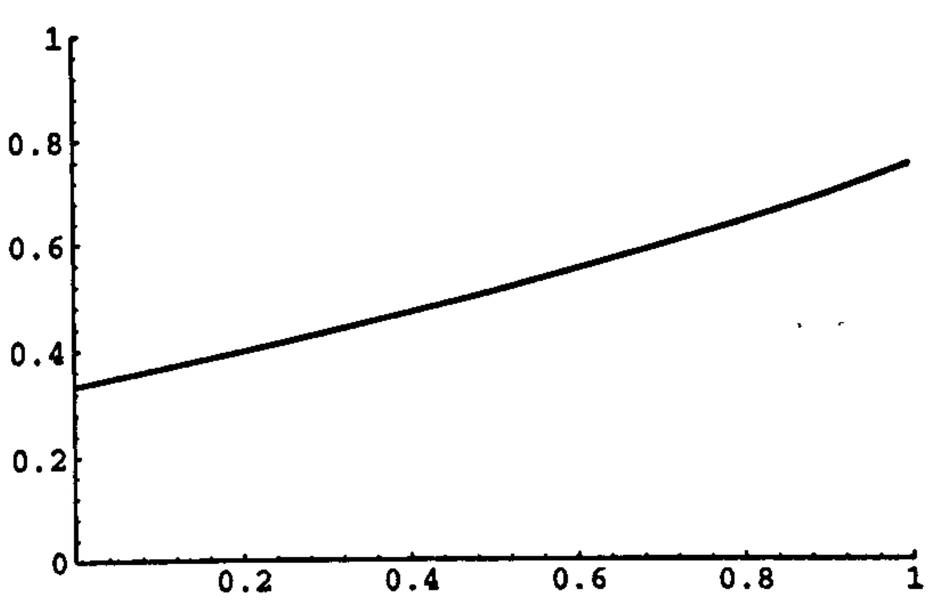

$A : P(A = sS)$



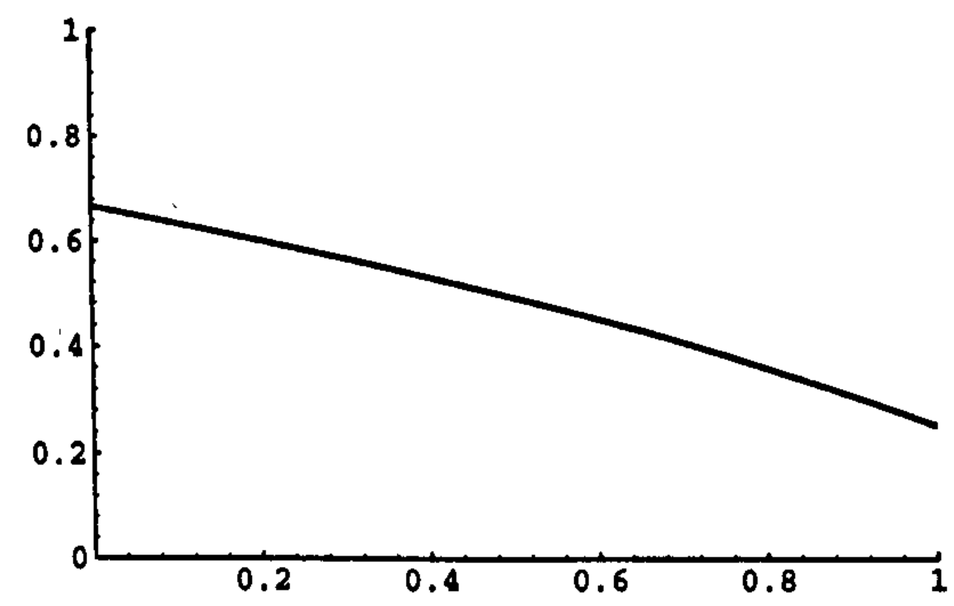

$B : P(B = ss)$



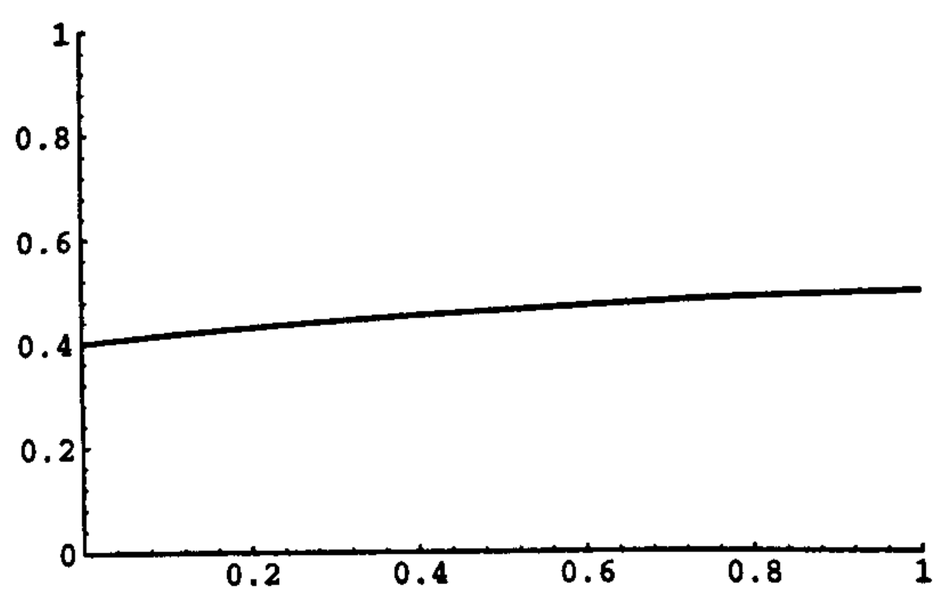

$B : P(B = sS)$



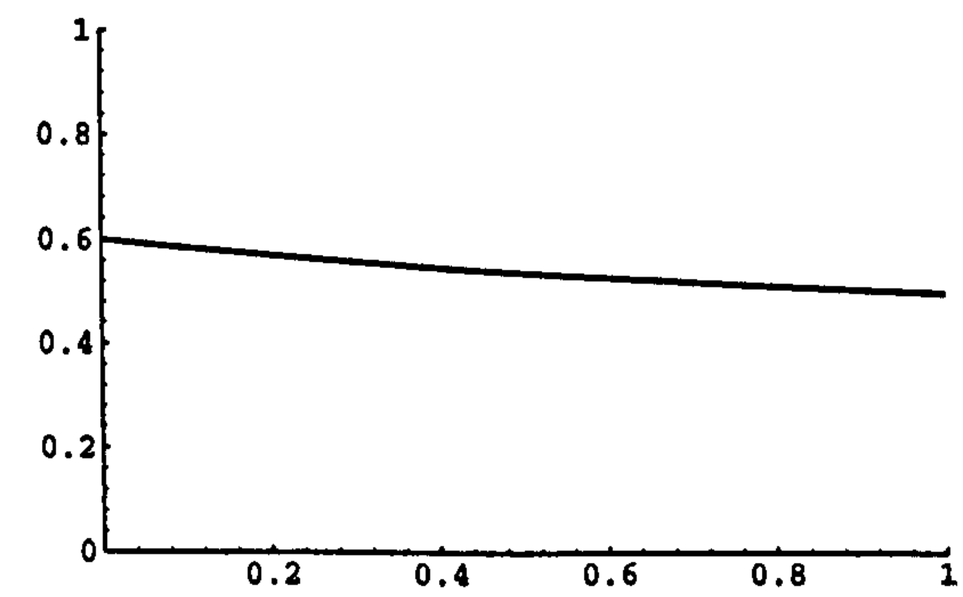

$C : P(C = ss)$




$C : P(C = sS)$

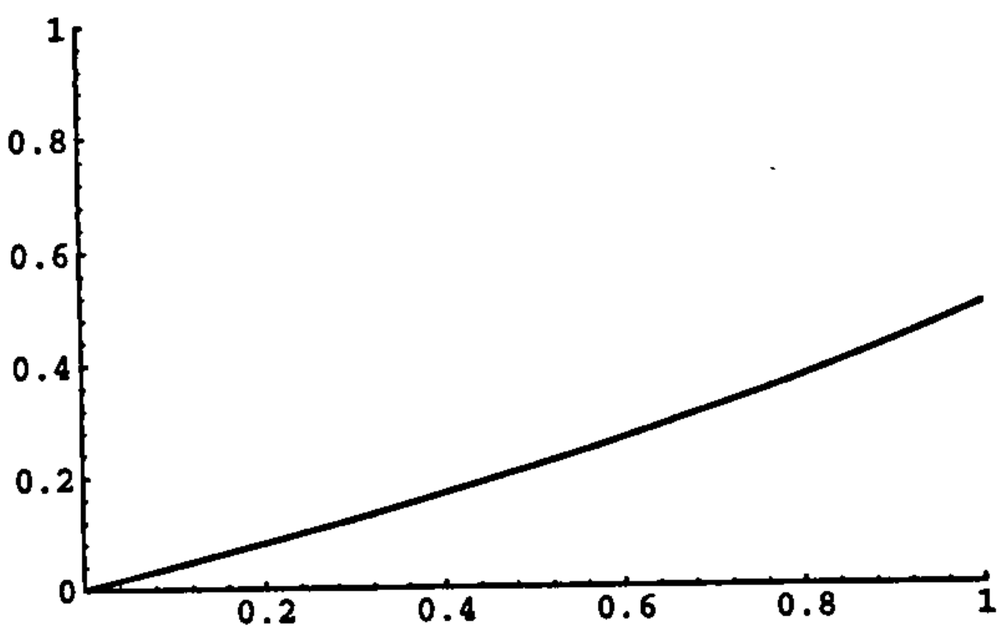



$D : P(D = ss)$

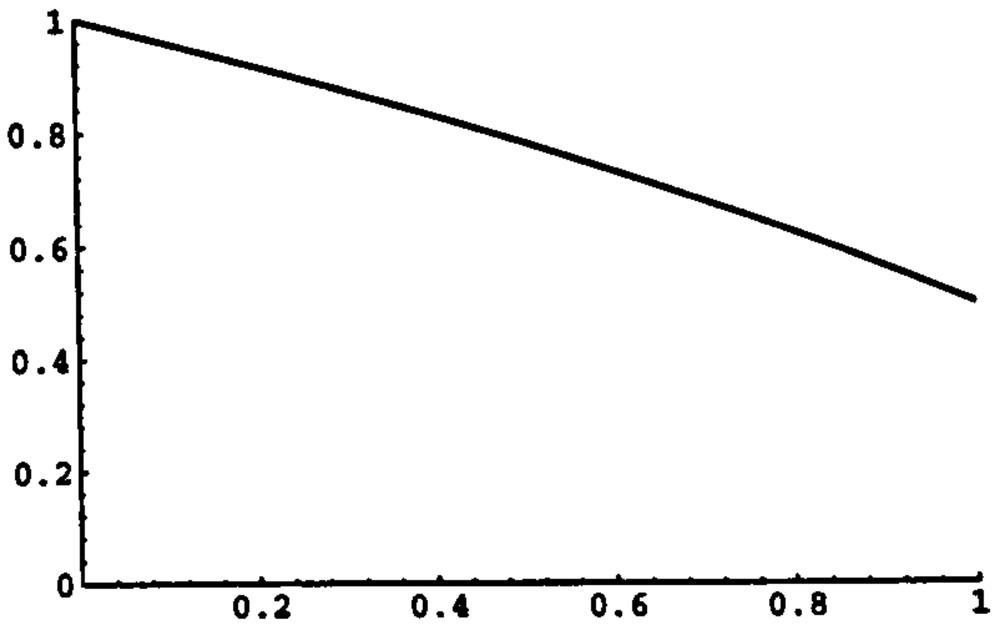



$D : P(D = sS)$

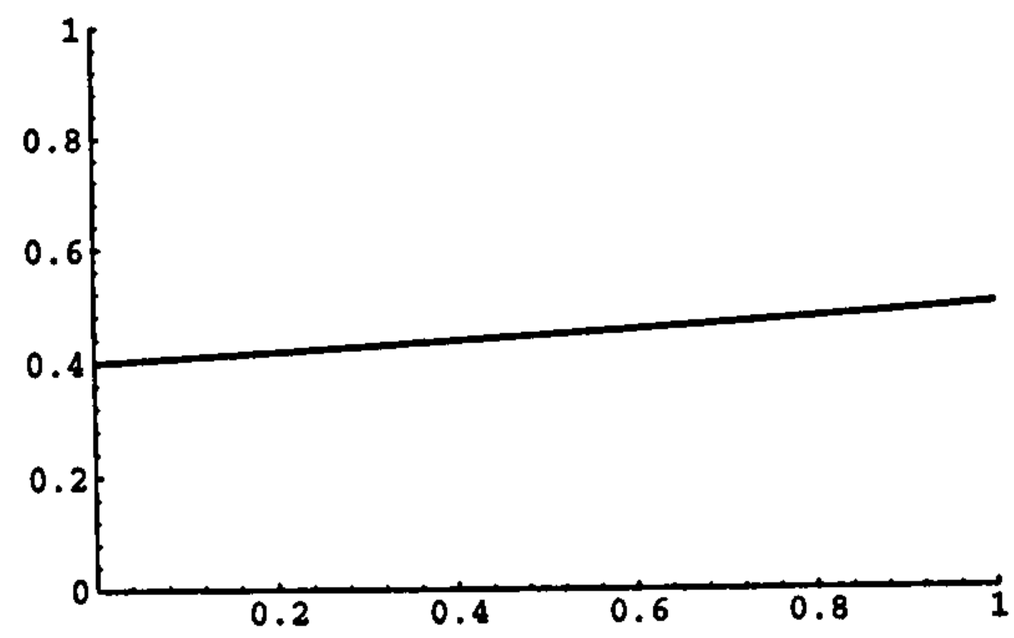
Figure 4.4: The marginal distributions of the genotypes $A$-$D$ in the genetics counselling problem for varying levels of $p$.



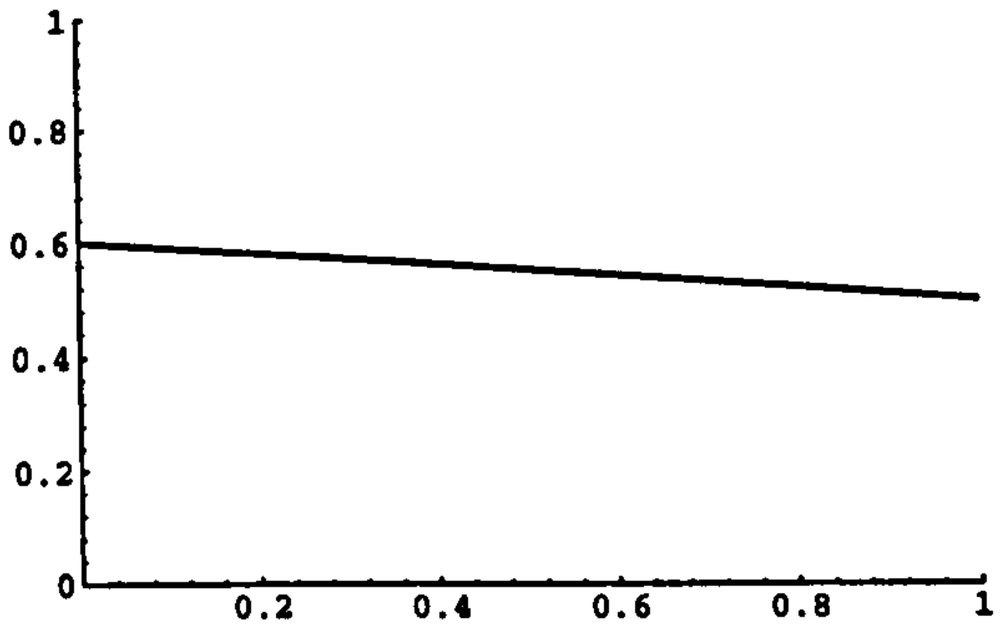

$E : P(E = ss)$



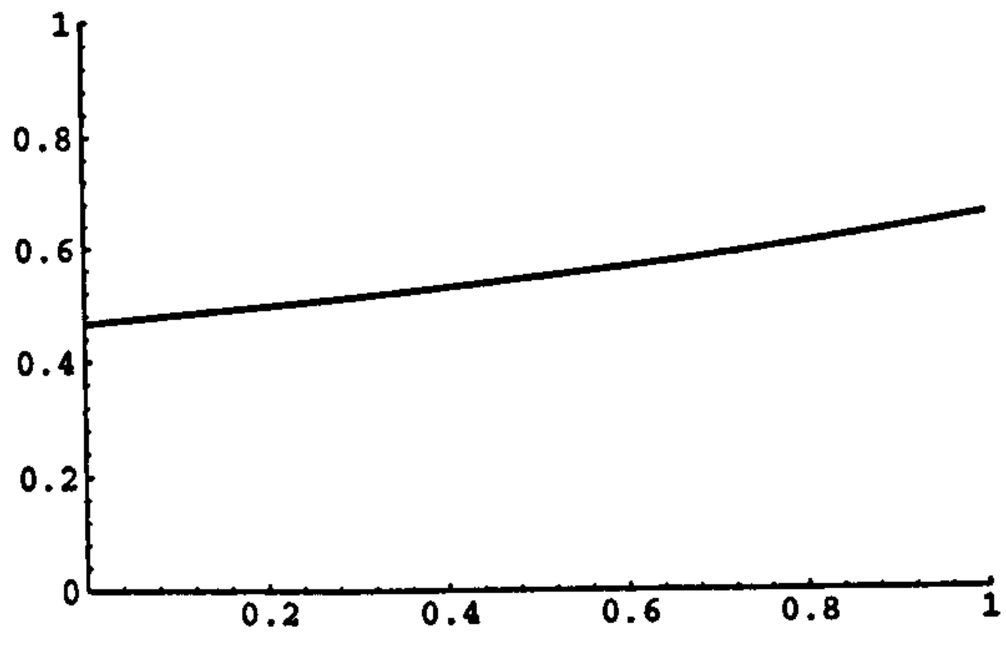

$E : P(E = sS)$



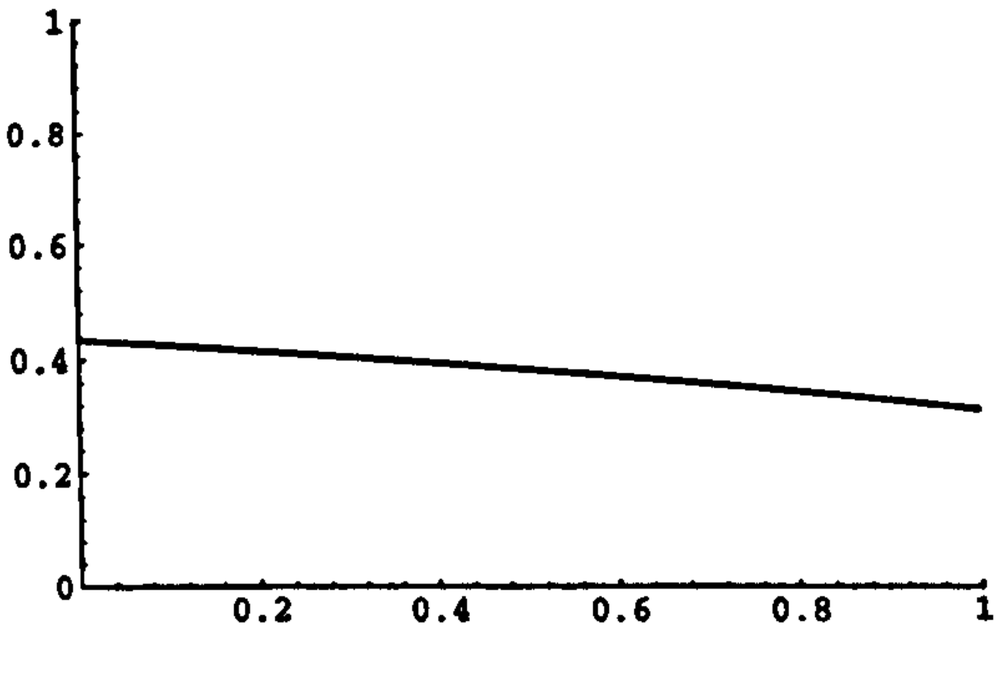

$F : P(F = ss)$



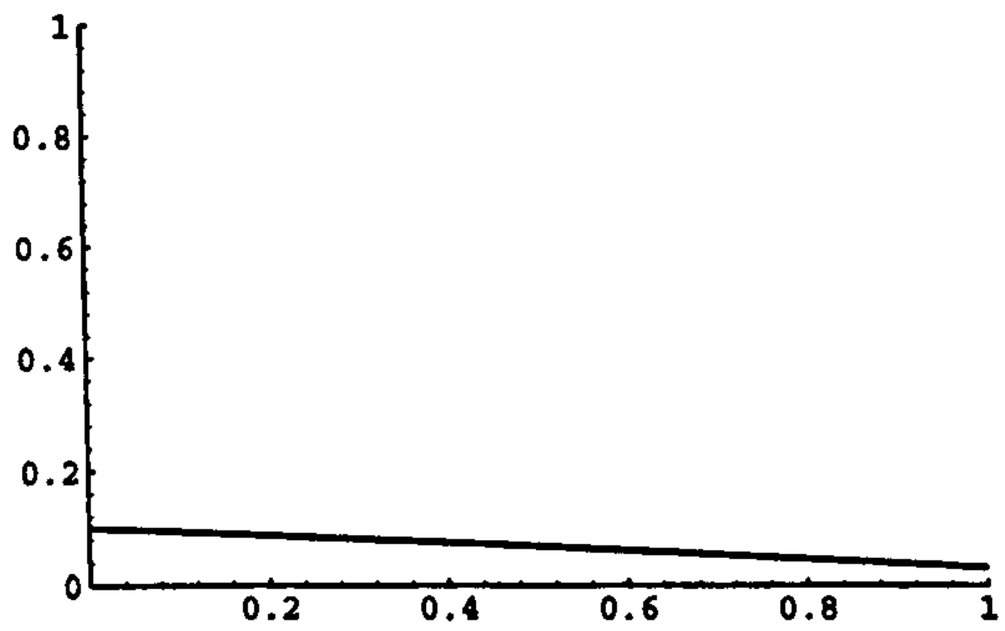

$F : P(F = sS)$




$I : P(I = ss)$

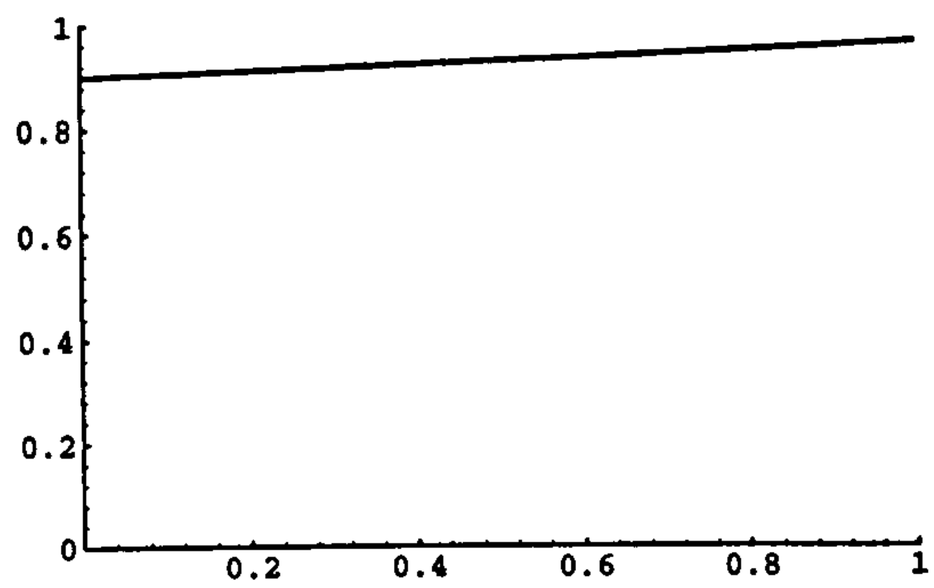



$I : P(I = sS)$

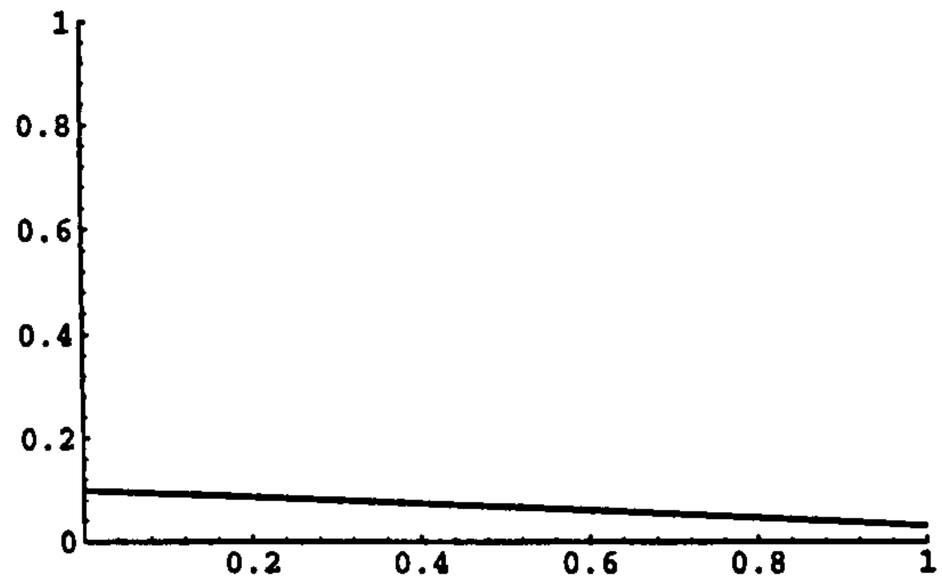



$I : P(I = SS)$

Figure 4.5: The marginal distributions of the genotypes $E$, $F$ and $I$ in the genetics counselling problem for varying levels of $p$.




$I' : P(I' = n)$




$I' : P(I' = \bar{n})$

Figure 4.6: The marginal distribution of the phenotype $I'$ in the genetics counselling problem for varying levels of $p$.

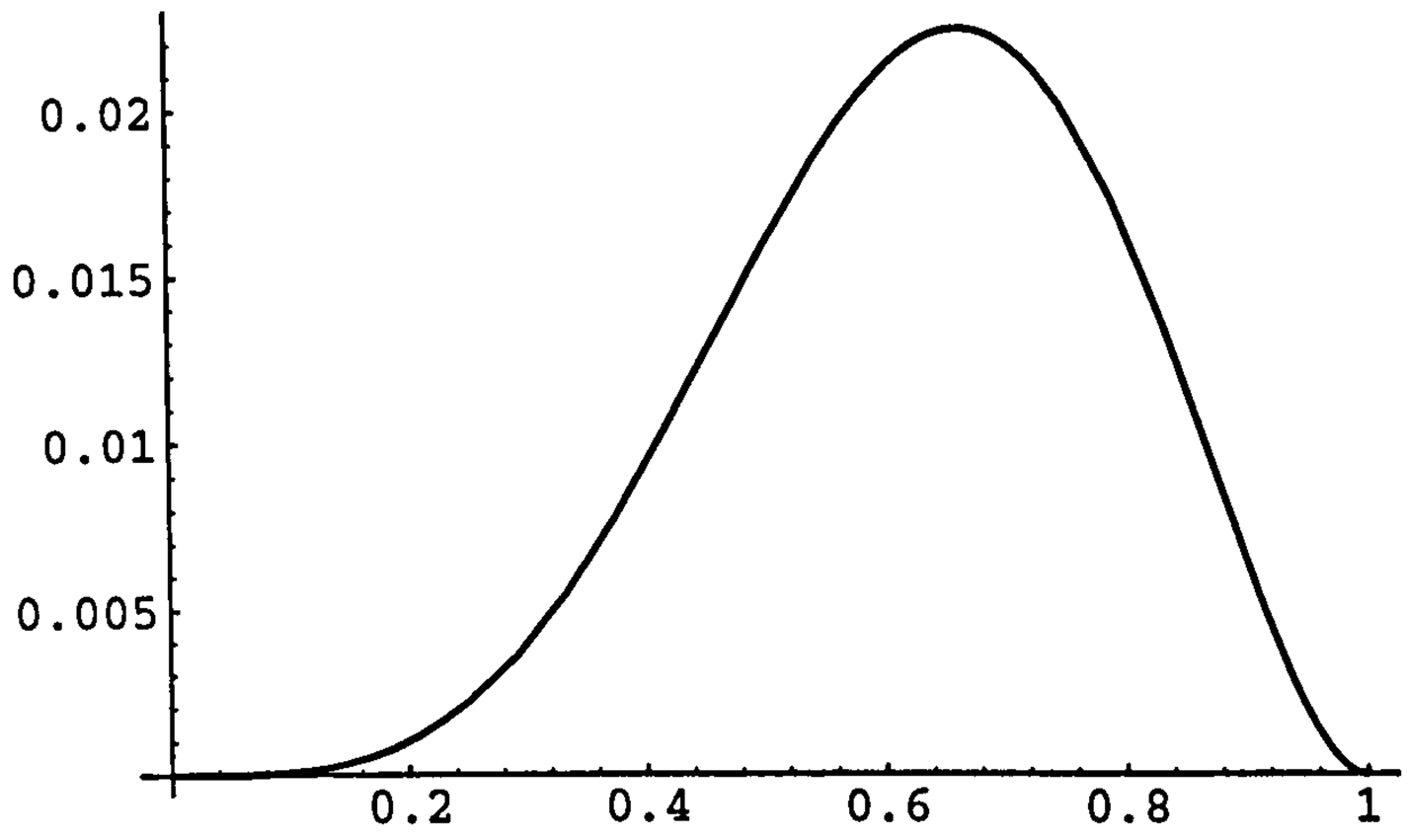



Figure 4.7: The normalisation constant in the genetics counselling problem for varying levels of $p$.

The normalisation constant as defined in Equation 4.1 is plotted for varying levels of $p$ in Figure 4.7. Recall that the normalisation constant is the probability of all the data observed, $P(A', \ldots, H', J')$, it is therefore the overall likelihood of the pedigree. We may thus determine the maximum likelihood of $p$ from the normalisation constant. We may differentiate Equation 4.1 with respect to $p$ in *Mathematica* using the `D` function. The first derivative of the normalisation constant with respect to $p$ is thus:

$$\frac{(p-1)p^3(3p-2)(15+5p-4p^2)}{8}$$

Setting this equation equal to zero we may use the `Solve` function in *Mathematica* to determine 5 distinct roots: $\frac{5-\sqrt{265}}{8}$, 0, $\frac{2}{3}$, 1, and $\frac{5+\sqrt{265}}{8}$. Since $0 \leq p \leq 1$,

by reference with Figure 4.7, the maximum likelihood estimate, $\hat{p}$, for $p$ is $\frac{2}{3}$. Alternatively we could use the second derivative of the normalisation constant to determine the appropriate root. The maximum likelihood estimates of the marginal distributions for the genotypes $A$-$J$, and the phenotype $I'$ are thus as given in Table 4.4.

**Genotypes:**

$P(X = x)$ **for** $X \in \{A, B, C, D, E, F, G, H, I, J\}$

| | x | | |
|---|---|---|---|
| $X$ | $ss$ | $sS$ | $SS$ |
| $A$ | $\frac{76}{147}$ | $\frac{71}{147}$ | 0 |
| $B$ | $\frac{76}{147}$ | $\frac{71}{147}$ | 0 |
| $C$ | $\frac{85}{147}$ | $\frac{62}{147}$ | 0 |
| $D$ | $\frac{10}{21}$ | $\frac{11}{21}$ | 0 |
| $E$ | $\frac{44}{147}$ | $\frac{103}{147}$ | 0 |
| $F$ | $\frac{68}{147}$ | $\frac{79}{147}$ | 0 |
| $G$ | 0 | 1 | 0 |
| $H$ | 0 | 1 | 0 |
| $I$ | $\frac{85}{147}$ | $\frac{107}{294}$ | $\frac{17}{294}$ |
| $J$ | 0 | 0 | 1 |

**Phenotype:**

$P(X = x)$ **for** $X \in \{I'\}$

| | x | |
|---|---|---|
| $X$ | $n$ | $\bar{n}$ |
| $I'$ | $\frac{277}{294}$ | $\frac{17}{294}$ |

Table 4.4: The maximum likelihood estimates of the marginal distributions on the genotypes $A$-$J$ and the phenotype $I'$ with $p = \hat{p} = \frac{2}{3}$.

Thus, following propagation and marginalisation, our symbolic methodology furnishes us not only with symbolic expressions from which we may immediately

derive the exact probabilities of the genotypic and phenotypic characteristics of every individual in the pedigree for any given population frequency $p$ of the recessive allele, but also a methodology for determining their maximum likelihood estimates. An efficient alternative approach using only numeric techniques is difficult to suggest and this will often be the case in genetics examples where many conditional distributions depend on a common parameter.

One possible solution might be to represent $p$ by an additional parent node $P$ in the causal probabilistic network with a suitable distribution. The problem here is that $P$ would need to be connected to the four founder nodes which would result in an inefficient increase in clique size. The distribution for $P$ would also be a problem, in order to increase accuracy and keep the problem numerically based a large number of values of $p$ should be allowed, each with equal probability. This would result in an inefficient increase in potential table size. The easiest solution might be to simply choose fixed values of $p$ and calculate the likelihood and probabilities of interest from them. Providing the likelihood is well-behaved, as in this case, it should not be too difficult to determine its maximum. With very large complicated pedigrees this latter approach might be the only course of action.

## 4.3 More Waste

We now turn our attention to consider how symbolic techniques may be used to model continuous random variables. This is an important area which, though it takes the most direct modelling route, seems to have previously received very little attention. The most likely reason for this is the relative inaccessibility of computer algebra. Only a few symbolic programming languages are available to the researcher, in contrast to the glut of numeric programming languages, and they are still very much in their infancy. *Mathematica*, for example, was only released in mid 1988. Such packages are also viewed by many as neat "toys" best suited to solving the odd differential equation, plotting a function or two, and keeping excessive algebra at bay. They are in their present state, however, excellent research languages appropriate for the construction of prototype probabilistic expert systems. These prototypes, once fully understood, may later be coded into dedicated systems which although reducing the systems flexibility should improve the performance of the system both in terms of the size of problem which may be modelled, and the speed with which computations are performed. The dedicated

system may also, by the addition of a suitable front end, be made less daunting to the non expert.

To consider the application of symbolic methods to systems with continuous variables we will return to Lauritzen's waste incinerator problem which we first introduced in Section 3.10. This problem seeks to model the processes in operation during the burning of waste in an incinerator plant. It is an example of a mixed graphical association model. It comprises a set of discrete random variables with finite state space of known dimension, and a set of continuous random variables which are conditionally normal given their parents. A restriction to the generality of the model exists that no continuous variable may have a discrete child.

Recall the general set up of mixed graphical association models. We have an independence graph $\mathcal{G} = (K, E)$ for a set of random variables, $X = (X_1, X_2, \ldots, X_k)$. $K$ is partitioned into a set of discrete vertices $\Delta$ and a set of continuous vertices $\Gamma$ such that $K = \Delta \cup \Gamma$. A typical element of the joint state space of discrete and continuous variables may be written in terms of its quantitative and qualitative components thus:

$$x = (x_a)_{a \in K} = (i, y) = ((i_\delta)_{\delta \in \Delta}, (y_\gamma)_{\gamma \in \Gamma})$$

where $i_\delta$ are qualitative values and $y_\gamma$ are real-valued. A combination of discrete variables $i = (i_\delta)_{\delta \in \Delta}$ is termed a *cell* and is akin to the cells of a contingency table formed by the qualitative variables.

A causal probabilistic network (CPN) is used to relate the conditional independence properties of the variables $X$ to a graph $\mathcal{G}$ with vertex set $K$. The CPN for the waste incinerator problem is given in Figure 4.8. A junction tree may be formed from the CPN by first applying a one-off compilation process consisting of moralisation and triangulation phases. In contrast to the procedure required by Lauritzen's scheme for the modelling of CG-potentials by use of their canonical and moment characteristics in our symbolic scheme we will not require the existence of a strong root. Therefore, the triangulation need only be weak, and hence the junction tree may only be *weakly decomposable*. A moral graph will always require less, or as many, links to be added to it to make it weakly rather than strongly triangulated. A weakly decomposable junction tree will therefore consist of clique and separator universes of lower or equal dimension to its strongly decomposable counterpart. This is a desirable feature of our technique since the

basic building blocks of the PES will be kept as small as possible. The weakly decomposable junction tree for the waste incinerator problem is given in Figure 4.9.

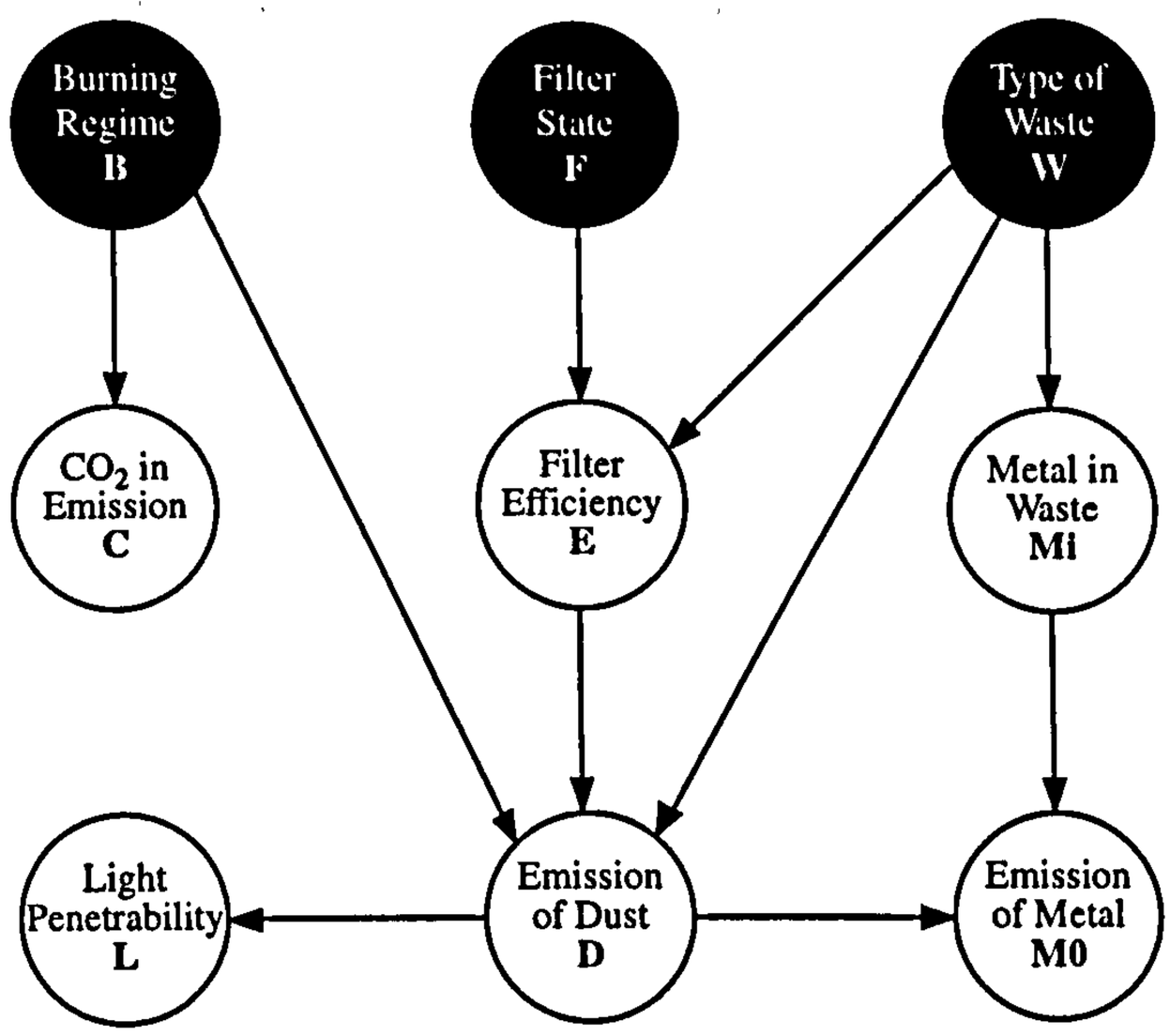



Figure 4.8: The causal probabilistic network of the Waste Incinerator Problem.

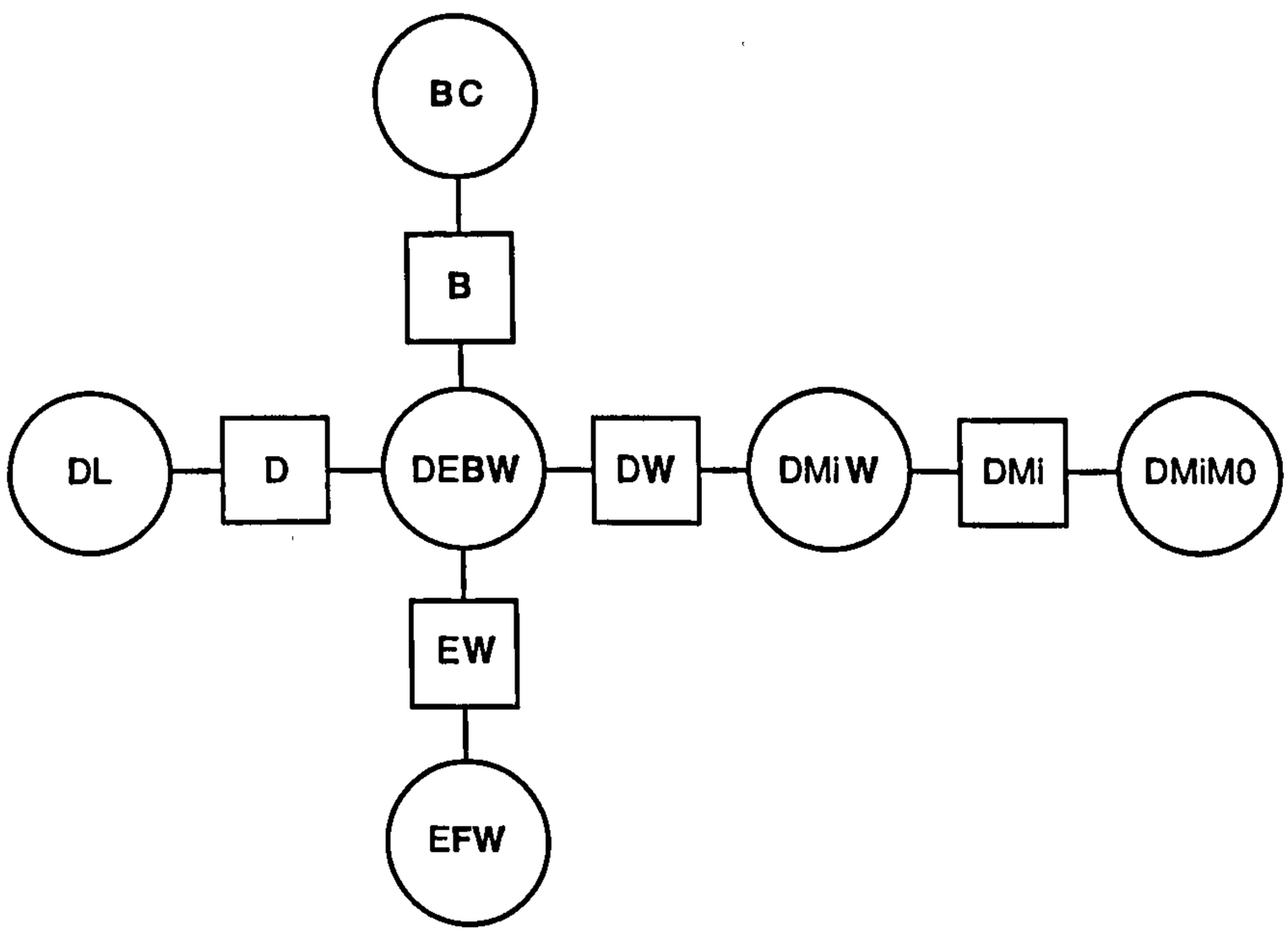



Figure 4.9: The weakly decomposable junction tree of the Waste Incinerator Problem.

We denote the junction tree as $\mathcal{T}$. It comprises a vertex-set $\mathcal{C}$ and an edge-set $\mathcal{S}$. $\mathcal{T}$ is assumed to be connected, since the disconnected case may be dealt with by separate PESs. Associated with any $C \in \mathcal{C}$ is a subset of $K$, which is denoted

by $C$ also. Any vertex $C$ is termed a *clique*. Associated with any $S \in \mathcal{S}$, joining two cliques $C$ and $C'$, is a subset of $K$ which we will denote by $S$ also, such that $S = C \cap C'$. Any edge $S$ is termed a *separator* since it separates two cliques.

## 4.4 CGM-Distributions and CGM-Potentials

We defined a CG-distribution in Definition 29. In this chapter we will be employing exact techniques to model the distributions in a mixed graphical association model. We will therefore need to generalise CG-distributions to allow for weighted mixtures of them. These we will term *conditional Gaussian mixture distributions* or *CGM-distributions* for short.

**Definition 35 CGM-Distribution:** *We term the joint density, $f_A$, of the variables $X_A$ for $A \subseteq K$ a* CGM-distribution, *if:*

$$f_A(x_A) = f_A(i_A, y_A) = \sum_{l=1}^{L} \chi_l(i_A)\, exp\{g_l(i_A) + h_l(i_A)^T y_A - \frac{1}{2} y_A^T J_l(i_A) y_A\} \quad (4.2)$$

*where $\chi_l(i_A) \in \{0,1\}$, for $l = 1,2,\ldots,L$, is an indicator function which controls the inclusion (or exclusion) of the $l-th$ term into the distribution. When $\chi_l(i_A) \equiv 0$ we need not explicitly define $g_l(i_A)$, $h_l(i_A)$ or $J_l(i_A)$.*

By Definitions 29 and 35 a CG-distribution is a special case of CGM-distribution for which $L \equiv 1$.

**Theorem 29** *Equation 4.2 is equivalent to the statement:*

$$f_{Y_A|(I_A=i_A)}(y_A; i_A) = \sum_{l=1}^{L} f_l(y_A; i_A) \quad whenever \quad p_l(i_A) > 0$$

*where $f_l(y_A; i_A)$ is the probability density function of a multivariate Normal distribution $N_{|\Gamma_A|}(\xi_l(i_A), \Sigma_l(i_A))$ and we define:*

$$\begin{aligned} p_l(i_A) &= (2\Pi)^{1/2|\Gamma_A|}\{det J_l(i_A)\}^{-1/2} exp\{g_l(i_A) + \frac{1}{2} h_l(i_A)^T J_l(i_A)^{-1} h_l(i_A)\} \\ \xi_l(i_A) &= J_l(i_A)^{-1} h_l(i_A) \\ \Sigma_l(i_A) &= J_l(i_A)^{-1}, \quad and \quad \Sigma_l \quad is\ positive\ definite. \end{aligned} \quad (4.3)$$

**Proof.** Using Equation 4.3 we may show that:

$$\begin{aligned}
f_A(x_A) &= f_A(i_A, y_A) \\
&= \sum_{l=1}^{L} \chi_l(i_A) \exp\left\{ g_l(i_A) + h_l(i_A)^T y_A - \frac{1}{2} y_A^T J_l(i_A) y_A \right\} \\
&= \sum_{l=1}^{L} \chi_l(i_A) \exp\left\{ \log p_l(i_A) + \frac{1}{2} \log \det \Sigma_l(i_A)^{-1} - \frac{1}{2} \mid \Gamma_A \mid \log(2\Pi) \right. \\
&\quad \left. - \frac{1}{2} \xi_l(i_A)^T \Sigma_l(i_A)^{-1} \xi_l(i_A) + \left( \Sigma_l(i_A)^{-1} \xi_l(i_A) \right)^T y_A - \frac{1}{2} y_A^T \Sigma_l(i_A)^{-1} y_A \right\} \\
&= \sum_{l=1}^{L} \chi_l(i_A) p_l(i_A) \{ \det \Sigma_l(i_A) \}^{-1/2} (2\Pi)^{-1/2|\Gamma_A|} \\
&\times \exp\left\{ -\frac{1}{2} \left( \xi_l(i_A)^T \Sigma_l(i_A)^{-1} \xi_l(i_A) - 2 \xi_l(i_A)^T \Sigma_l(i_A)^{-1} y_A \right. \right. \\
&\quad \left. \left. + y_A^T \Sigma_l(i_A)^{-1} y_A \right) \right\} \\
&= \sum_{l=1}^{L} \chi_l(i_A) p_l(i_A) \{ \det \Sigma_l(i_A) \}^{-1/2} (2\Pi)^{-1/2|\Gamma_A|} \\
&\times \exp\left\{ -\frac{1}{2} (y_A - \xi_l(i_A))^T \Sigma_l(i_A)^{-1} (y_A - \xi_l(i_A)) \right\} \\
&= \sum_{l=1}^{L} \chi_l(i_A) p_l(i_A) f_l(y_A; i_A)
\end{aligned}$$

Thus we see that the conditional distribution of the set of quantitative variables, $Y_A$, given the qualitative variables, $I_A$, is a weighted mixture of multivariate Gaussian distributions. We may also note that:

$$P(I_A = i_A) = \sum_{l=1}^{L} \chi_l(i_A) p_l(i_A)$$

□

In order to define a CGM-distribution uniquely we may therefore quote one of two possible sets of triples - the *canonical characteristics* $(g_l, h_l, J_l)$ for $l = 1, 2, \ldots, L$ or the *moment characteristics* $(p_l, \xi_l, \Sigma_l)$ for $l = 1, 2, \ldots, L$. We note that we may convert from the canonical characteristics to the moment characteristics using Equations 4.3, and from the moment characteristics to the canonical characteristics using Equations 4.4. Such conversions will not, however, be necessary for our scheme.

$$\begin{aligned}
g_l(i_A) &= \log p_l(i_A) + \frac{1}{2} \{ \log \det \Sigma_l(i_A)^{-1} - \mid \Gamma_A \mid \log(2\Pi) - \xi_l(i_A)^T \Sigma_l(i_A)^{-1} \xi_l(i_A) \} \\
h_l(i_A) &= \Sigma_l(i_A)^{-1} \xi_l(i_A) \\
J_l(i_A) &= \Sigma_l(i_A)^{-1}
\end{aligned} \tag{4.4}$$

A more general representation of a CGM-distribution for a set of random variables $X_A$, $A \subseteq K$, is any function $\phi_A$ of the form:

$$\phi_A(x_A) = \phi_A(i_A, y_A) = \sum_{l=1}^{L} \chi_l(i_A) \exp\{g_l(i_A) + h_l(i_A)^T y_A - \frac{1}{2} y_A^T J_l(i_A) y_A\} \quad (4.5)$$

where $J_l(i_A)$ is symmetric and $\phi_A$ is not necessarily a density. We term $\phi_A$ a *CGM-potential*. The conversion formulae of Equations 4.3 and 4.4 apply equally to CGM-potentials. If $L \equiv 1$ Equation 4.5 defines a CG-potential.

## 4.5 Defining the Variables

Let us assume that in our CPN no node which relates to a continuous random variable may have a child which relates to a discrete random variable and that every discrete random variable may be constrained to a known finite set of levels. Let $a \in \Delta$ be a node in the CPN with associated discrete random variable $I_a$. Let $I_a$ have $n_a$ levels then we may define its conditional distribution, given its parents $X_{pa(a)} = I_{pa(a)}$, which are then all ensured to be discrete, as follows:

$$f_{a|pa(a)}\left(i_a; i_{pa(a)}\right) = P(I_a = i_a \mid I_{pa(a)} = i_{pa(a)}) = p(i_a; i_{pa(a)}) \quad (4.6)$$

We will assume that the continuous random variables in our system are conditionally Gaussian. Let $a \in \Gamma$ be a node in the CPN with associated continuous random variable $Y_a$. Then the conditional distribution of $Y_a$ given its parents $X_{pa(a)}$ is of the following type:

$$Y_a \mid X_{pa(a)} \sim N\left(\alpha(i_{pa(a)}) + \beta(i_{pa(a)})^T y_{pa(a)}, \gamma(i_{pa(a)})\right) \quad (4.7)$$

where $X_{pa(a)}$ has state space $\mathcal{I}_{pa(a)} \times I\!R^{|\Gamma_{pa(a)}|}$, $\alpha(i_{pa(a)}) \in I\!R$, $\beta(i_{pa(a)}) \in I\!R^{|\Gamma_{pa(a)}|}$ and $\gamma(i_{pa(a)}) > 0$. Thus the mean of the conditional distribution of $Y_a$ given its parents is a linear function of the states at the continuous parent nodes, $Y_{pa(a)}$, while the variance does not depend on the continuous parent nodes at all. Both mean and variance may depend on the discrete parents.

The marginal and conditional distributions of the random variables in the waste incinerator problem are given in Table 4.5. The three discrete variables $B$, $F$, and $W$ are founders and are thus defined as marginal distributions. The continuous variables are all non-founders, however, and are thus defined as conditional distributions. The distributions in Table 4.5 are presented in a way that is both simple to code into *Mathematica* and, since all numbers are given as fractions, will be ensured to maintain maximum accuracy.

| B: Burning Regime | | F: Filter State | | W: Type of Waste | |
|---|---|---|---|---|---|
| $b$ | $P(B = b)$ | $f$ | $P(F = f)$ | $w$ | $P(W = w)$ |
| Stable | 17/20 | Intact | 19/20 | Industrial | 2/7 |
| Unstable | 3/20 | Defective | 1/20 | Household | 5/7 |

| C \| B (C: C02 in Emission) | | Mi \| W (Mi: Metal in Waste) | |
|---|---|---|---|
| B: Burning Regime | | W: Type of Waste | |
| Stable | N(-2, 1/10) | Industrial | N(1/2, 1/100) |
| Unstable | N(-1, 3/10) | Household | N(-1/2, 1/200) |

| E \| (F, W) (E: Filter Efficiency) | | |
|---|---|---|
| F: Filter State | W: Type of Waste | |
| | Industrial | Household |
| Intact | N(-39/10, 1/50000) | N(-16/5, 1/50000) |
| Defective | N(-2/5, 1/10000) | N(-1/2, 1/10000) |

| D \| (B, E, W) (D: Emission of Dust, E: Filter Efficiency) | | |
|---|---|---|
| B: Burning Regime | W: Type of Waste | |
| | Industrial | Household |
| Stable | N(13/2 + e, 3/100) | N(6 + e, 1/25) |
| Unstable | N(15/2 + e, 1/10) | N(7 + e, 1/10) |

| L \| D (L: Light Penetrability, D: Emission of Dust) | M0 \| D, Mi (M0: Emission of Metal, D: Emission of Dust, Mi: Metal in Waste |
|---|---|
| N(3 - 1/2 d, 1/4) | N(d + mi, 1/500) |

Table 4.5: The marginal and conditional distributions of the variables in the waste incinerator problem.

## 4.6 Basic Operations

In this section we shall consider how CGM-potentials are affected by the basic operations required to pass an absorption in a PES. These basic operations are *extension*, *multiplication*, *division* and *marginalisation*.

### 4.6.1 Extending a CGM-Potential

**Definition 36 Extension of a CGM-Potential:** *Let $U \subseteq V \subseteq K$ and $\phi(x_U) = \phi(i_U, y_U)$ be a CGM-potential defined on $\mathcal{X}_U = \mathcal{I}_U \times \mathcal{Y}_U$. We define $\eta(x_V) = \eta(i_V, y_V)$ to be the* **extension** *of $\phi(x_U)$ to $V$, where $\eta(x_V)$ is a CGM-potential defined on $\mathcal{X}_V = \mathcal{I}_V \times \mathcal{Y}_V = (\mathcal{I}_U \times \mathcal{I}_{V\setminus U}) \times (\mathcal{Y}_U \times \mathcal{Y}_{V\setminus U})$ and let:*

$$\eta(x_V) = \eta(i_U, i_{V\setminus U}, y_U, y_{V\setminus U}) = \phi(i_U, y_U) \tag{4.8}$$

Thus the extension of a CGM-potential $\phi(x_U)$ from $\mathcal{X}_U$ to $\mathcal{X}_V$ simply duplicates the potential over the discrete space $\mathcal{I}_{V\setminus U}$. This simplifies the application of other operators and has no effect on the distributional properties of the potential. The extension is therefore still a CGM-potential.

### 4.6.2 Multiplying Two CGM-Potentials

**Definition 37 Multiplication of two CGM-Potentials:** *Let $\phi$ and $\eta$ be two CGM-potentials defined on the spaces $\mathcal{X}_U = \mathcal{I}_U \times \mathcal{Y}_U$ and $\mathcal{X}_V = \mathcal{I}_V \times \mathcal{Y}_V$ respectively with $U \subseteq K$ and $V \subseteq K$. The* **multiplication** *of $\phi$ and $\eta$, denoted $\phi \times \eta$, defined on the space $\mathcal{X}_{U\cup V} = \mathcal{I}_{U\cup V} \times \mathcal{Y}_{U\cup V}$ is defined to be:*

$$(\phi \times \eta)(x_{U\cup V}) = \phi(x_{U\cup V})\eta(x_{U\cup V}) \tag{4.9}$$

*where $\phi$ and $\eta$ on the right-hand side have first been extended to occupy $\mathcal{X}_{U\cup V}$.*

**Theorem 30** *Let $\phi$ and $\eta$ be two CGM-potentials extended to occupy the same space $\mathcal{X}_V = \mathcal{I}_V \times \mathcal{Y}_V$ where $V \subseteq K$. Then their product, $\phi \times \eta$, is a CGM-potential also.*

**Proof.**

$$\begin{aligned}
\phi(x_V) \times \eta(x_V) &= \sum_{l=1}^{L} \chi_l(iv) \exp\left\{g_l(iv) + h_l(iv)^T y_V - \frac{1}{2} y_V^T J_l(iv) y_V\right\} \\
&\times \sum_{m=1}^{M} \chi_m^*(iv) \exp\left\{g_m^*(iv) + h_m^*(iv)^T y_V - \frac{1}{2} y_V^T J_m^*(iv) y_V\right\} \\
&= \sum_{l=1}^{L} \sum_{m=1}^{M} \Big(\chi_l(iv) \times \chi_m^*(iv)\Big) \exp\left\{\Big(g_l(iv) + g_m^*(iv)\Big)\right. \\
&+ \left.\Big(h_l(iv) + h_m^*(iv)\Big)^T y_V - \frac{1}{2} y_V^T \Big(J_l(iv) + J_m^*(iv)\Big) y_V\right\} \\
&= \sum_{l'=1}^{L'} \chi'_{l'}(iv) \exp\left\{g'_{l'}(iv) + h'_{l'}(iv)^T y_V - \frac{1}{2} y_V^T J'_{l'}(iv) y_V\right\}
\end{aligned}$$

Thus the product forms a CGM-potential.

□

### 4.6.3 Dividing Two CGM-Potentials

**Definition 38 Division of two CGM-Potentials:** *Let $\phi$ and $\eta$ be two CGM-potentials defined on the spaces $\mathcal{X}_U = \mathcal{I}_U \times \mathcal{Y}_U$ and $\mathcal{X}_V = \mathcal{I}_V \times \mathcal{Y}_V$ respectively with $U \subseteq K$ and $V \subseteq K$. The* **division** *of $\phi$ by $\eta$, denoted $\phi/\eta$, defined on the space $\mathcal{X}_{U\cup V} = \mathcal{I}_{U\cup V} \times \mathcal{Y}_{U\cup V}$ is defined to be:*

$$(\phi/\eta)(x_{U\cup V}) = \begin{cases} 0 & if\, \phi(x_{U\cup V}) = 0 \\ (\phi(x_{U\cup V})/\eta(x_{U\cup V})) & if\, \eta(x_{U\cup V}) \neq 0 \\ undefined & otherwise \end{cases} \tag{4.10}$$

*where $\phi$ and $\eta$ on the right-hand side have first been extended to occupy $\mathcal{X}_{U\cup V}$.*

**Theorem 31** *Let $\phi$ and $\eta$ be two CGM-potentials extended to occupy the same space $\mathcal{X}_V = \mathcal{I}_V \times \mathcal{Y}_V$ where $V \subseteq K$ and let $\phi(x_V) \neq 0$ and $\eta(x_V) \neq 0$. Suppose also that either $\eta \equiv 1$ or $\phi = \eta \times \phi'$, where $\phi'$ is a CGM-potential. Then $\phi/\eta$, the division of $\phi$ by $\eta$, is a CGM-potential also.*

**Proof.** If $\eta \equiv 1$ then $\phi/\eta = \phi/1 = \phi$ but since $\phi$ is a CGM-potential then $\phi/\eta$ is a CGM-potential also. Suppose, however, that $\phi = \eta \times \phi'$ then $\phi/\eta = (\eta \times \phi')/\eta = \phi'$ but since $\phi'$ is a CGM-potential then $\phi/\eta$ is a CGM-potential also.

□

### 4.6.4 Marginalising over Discrete Variables

**Definition 39 Discrete Marginalisation of a CGM-Potential:** *Let* $U \subseteq V \subseteq K$ *and let* $\phi(x_V) = \phi(i_V, y_V) = \phi(i_U, i_{V\setminus U}, y_V)$ *be a CGM-potential defined on* $\mathcal{X}_V = \mathcal{I}_V \times \mathcal{Y}_V = (\mathcal{I}_U \times \mathcal{I}_{V\setminus U}) \times \mathcal{Y}_V$. *Then we will let the expression* $\sum_{V\setminus U} \phi(x_V)$ *denote the* **marginalisation** *of* $\phi(x_V)$, *with respect to* $I_{V\setminus U}$, *to a function* $\eta(x_U)$ *defined on the space* $\mathcal{X}_U = \mathcal{I}_U \times \mathcal{Y}_V$ *where:*

$$\eta(x_U) = \sum_{V\setminus U} \phi(x_V) = \sum_{i_{V\setminus U}} \phi(x_V) \tag{4.11}$$

**Theorem 32** *Let* $U \subseteq V \subseteq K$ *and* $\phi(x_V) = \phi(i_V, y_V) = \phi(i_U, i_{V\setminus U}, y_V)$ *be a CGM-potential defined on* $\mathcal{X}_V = \mathcal{I}_V \times \mathcal{Y}_V = (\mathcal{I}_U \times \mathcal{I}_{V\setminus U}) \times \mathcal{Y}_V$. *Let* $\eta(x_U) = \eta(i_U, y_V) = \sum_{i_{V\setminus U}} \phi(x_V)$ *be the marginal of* $\phi(x_V)$ *with respect to* $I_{V\setminus U}$. *Then the marginal* $\eta(x_U)$ *will be a CGM-potential also.*

**Proof.**

$$\begin{aligned}
\eta(i_U, y_V) &= \sum_{i_{V\setminus U}} \phi(i_U, i_{V\setminus U}, y_V) \\
&= \sum_{i_{V\setminus U}} \sum_{l=1}^{L} \chi_l(i_U, i_{V\setminus U}) \exp\left\{ g_l(i_U, i_{V\setminus U}) + h_l(i_U, i_{V\setminus U})^T y_V \right. \\
&\quad \left. -\frac{1}{2} y_V^T J_l(i_U, i_{V\setminus U}) y_V \right\} \\
&= \sum_{m=1}^{M} \chi'_m(i_U) \exp\left\{ g'_m(i_U) + h'_m(i_U)^T y_V - \frac{1}{2} y_V^T J'_m(i_U) y_V \right\}
\end{aligned}$$

Thus $\eta(i_U, y_V)$ is a CGM-potential also.

□

### 4.6.5 Marginalising over Continuous Variables

**Definition 40 Continuous Marginalisation of a CGM-Potential:** *Let* $U \subseteq V \subseteq K$ *and let* $\phi(x_V) = \phi(i_V, y_V) = \phi(i_V, y_{V\setminus U}, y_U)$ *be a CGM-potential defined on* $\mathcal{X}_V = \mathcal{I}_V \times \mathcal{Y}_V = \mathcal{I}_V \times (\mathcal{Y}_{V\setminus U} \times \mathcal{Y}_U)$. *Then we will let the expression* $\sum_{V\setminus U} \phi(x_V)$ *denote the* **marginalisation** *of* $\phi(x_V)$, *with respect to* $Y_{V\setminus U}$, *to a function* $\eta(x_U)$ *defined on the space* $\mathcal{X}_U = \mathcal{I}_V \times \mathcal{Y}_U$ *where:*

$$\eta(x_U) = \sum_{V\setminus U} \phi(x_V) = \int_{y_{V\setminus U}=-\infty}^{y_{V\setminus U}=+\infty} \phi(x_V)\, dy_{V\setminus U} \tag{4.12}$$

**Theorem 33** *Let* $U \subseteq V \subseteq K$ *and* $\phi(x_V) = \phi(i_V, y_V) = \phi(i_V, y_{V\setminus U}, y_U)$ *be a CGM-potential defined on* $\mathcal{X}_V = \mathcal{I}_V \times \mathcal{Y}_V = \mathcal{I}_V \times (\mathcal{Y}_{V\setminus U} \times \mathcal{Y}_U)$ *such that:*

$$\phi(x_V) = \sum_{l=1}^{L} \chi_l(i_V)\ exp\left\{g_l(i_V) + h_l(i_V)^T y_V - \frac{1}{2} y_V^T J_l(i_V) y_V\right\} \tag{4.13}$$

*We may partition* $y_V$, *and* $h_l$ *and* $J_l$ *for* $l = 1, 2, \ldots, L$ *as:*

$$y_V = \begin{pmatrix} y_{V\setminus U} \\ y_U \end{pmatrix}\ h_l(i_V) = \begin{pmatrix} h^l_{V\setminus U}(i_V) \\ h^l_U(i_V) \end{pmatrix}\ J_l(i_V) = \begin{pmatrix} J^l_{V\setminus U, V\setminus U}(i_V) & J^l_{V\setminus U, U}(i_V) \\ J^l_{U, V\setminus U}(i_V) & J^l_{U,U}(i_V) \end{pmatrix}$$

*Then the marginalisation of* $\phi(x_V)$ *with respect to* $Y_{V\setminus U}$ *is a CGM-potential* $\eta(x_U) = \eta(i_V, y_U)$ *also.*

**Proof.** We may express $\phi(i_V, y_{V\setminus U}, y_U)$ as follows:

$$\begin{aligned}
&\phi(i_V, y_{V\setminus U}, y_U) \\
&= \sum_{l=1}^{L} \chi_l(i_V) \exp\Big\{g_l(i_V) + h_l(i_V)^T y_V - \frac{1}{2} y_V^T J_l(i_V) y_V\Big\} \\
&= \sum_{l=1}^{L} \chi_l(i_V) \exp\Big\{g_l(i_V) + \begin{pmatrix} h^l_{V\setminus U}(i_V)^T & h^l_U(i_V)^T \end{pmatrix} \begin{pmatrix} y_{V\setminus U} \\ y_U \end{pmatrix} \\
&\quad - \frac{1}{2} \begin{pmatrix} y^T_{V\setminus U} & y^T_U \end{pmatrix} \begin{pmatrix} J^l_{V\setminus U, V\setminus U}(i_V) & J^l_{V\setminus U, U}(i_V) \\ J^l_{U, V\setminus U}(i_V) & J^l_{U,U}(i_V) \end{pmatrix} \begin{pmatrix} y_{V\setminus U} \\ y_U \end{pmatrix}\Big\} \\
&= \sum_{l=1}^{L} \chi_l(i_V) \exp\Big\{g_l(i_V) + h^l_{V\setminus U}(i_V)^T y_{V\setminus U} + h^l_U(i_V)^T y_U \\
&\quad - \frac{1}{2}\Big(y^T_{V\setminus U} J^l_{V\setminus U, V\setminus U}(i_V) y_{V\setminus U} + y^T_{V\setminus U} J^l_{V\setminus U, U}(i_V) y_U \\
&\quad + y^T_U J^l_{U, V\setminus U}(i_V) y_{V\setminus U} + y^T_U J^l_{U,U}(i_V) y_U\Big)\Big\} \\
&= \sum_{l=1}^{L} \chi_l(i_V) \exp\Big\{ - \frac{1}{2}\Big(y^T_{V\setminus U} J^l_{V\setminus U, V\setminus U}(i_V) y_{V\setminus U} + y^T_{V\setminus U} J^l_{V\setminus U, U}(i_V) y_U \\
&\quad + y^T_U J^l_{U, V\setminus U}(i_V) y_{V\setminus U} - 2h^l_{V\setminus U}(i_V)^T y_{V\setminus U}\Big) \\
&\quad + g_l(i_V) + h^l_U(i_V)^T y_U - \frac{1}{2} y^T_U J^l_{U,U}(i_V) y_U\Big\} \\
&= \sum_{l=1}^{L} \chi_l(i_V) \exp\Big\{ - \frac{1}{2}\Big(y^T_{V\setminus U} + y^T_U J^l_{U, V\setminus U}(i_V) J^l_{V\setminus U, V\setminus U}(i_V)^{-1} \\
&\quad - h^l_{V\setminus U}(i_V)^T J^l_{V\setminus U, V\setminus U}(i_V)^{-1}\Big) J^l_{V\setminus U, V\setminus U}(i_V) \\
&\quad \Big(y_{V\setminus U} + J^l_{V\setminus U, V\setminus U}(i_V)^{-1} J^l_{V\setminus U, U}(i_V) y_U - J^l_{V\setminus U, V\setminus U}(i_V)^{-1} h^l_{V\setminus U}(i_V)\Big)
\end{aligned}$$

$$
\begin{aligned}
&+\frac{1}{2}\Big(y_U^T J^l_{U,V\setminus U}(iv) J^l_{V\setminus U,V\setminus U}(iv)^{-1} J^l_{V\setminus U,U}(iv) y_U \\
&-y_U^T J^l_{U,V\setminus U}(iv) J^l_{V\setminus U,V\setminus U}(iv)^{-1} h^l_{V\setminus U}(iv) \\
&-h^l_{V\setminus U}(iv)^T J^l_{V\setminus U,V\setminus U}(iv)^{-1} J^l_{V\setminus U,U}(iv) y_U \\
&+h^l_{V\setminus U}(iv)^T J^l_{V\setminus U,V\setminus U}(iv)^{-1} h^l_{V\setminus U}(iv)\Big) \\
&+g_l(iv)+h^l_U(iv)^T y_U-\frac{1}{2}y_U^T J^l_{U,U}(iv) y_U\Big\} \\
=\ &\sum_{l=1}^{L}\chi_l(iv)\exp\Big\{-\frac{1}{2}\Big(y_{V\setminus U}+J^l_{V\setminus U,V\setminus U}(iv)^{-1}J^l_{V\setminus U,U}(iv)y_U \\
&-J^l_{V\setminus U,V\setminus U}(iv)^{-1}h^l_{V\setminus U}(iv)\Big)^T J^l_{V\setminus U,V\setminus U}(iv) \\
&\Big(y_{V\setminus U}+J^l_{V\setminus U,V\setminus U}(iv)^{-1}J^l_{V\setminus U,U}(iv)y_U-J^l_{V\setminus U,V\setminus U}(iv)^{-1}h^l_{V\setminus U}(iv)\Big) \\
&+g_l(iv)+\frac{1}{2}h^l_{V\setminus U}(iv)^T J^l_{V\setminus U,V\setminus U}(iv)^{-1}h^l_{V\setminus U}(iv) \\
&+\Big(h^l_U(iv)-J^l_{U,V\setminus U}(iv)J^l_{V\setminus U,V\setminus U}(iv)^{-1}h^l_{V\setminus U}(iv)\Big)^T y_U \\
&-\frac{1}{2}y_U^T\Big(J^l_{U,U}(iv)-J^l_{U,V\setminus U}(iv)J^l_{V\setminus U,V\setminus U}(iv)^{-1}J^l_{V\setminus U,U}(iv)\Big)y_U\Big\}
\end{aligned}
\tag{4.14}
$$

Thus using Equation 4.14:

$$
\begin{aligned}
&\int_{y_{V\setminus U}=-\infty}^{y_{V\setminus U}=+\infty}\phi(iv,y_{V\setminus U},y_U)\,dy_{V\setminus U} \\
&=\sum_{l=1}^{L}\chi_l(iv)\exp\Big\{g_l(iv)+\frac{1}{2}h^l_{V\setminus U}(iv)^T J^l_{V\setminus U,V\setminus U}(iv)^{-1}h^l_{V\setminus U}(iv) \\
&+\Big(h^l_U(iv)-J^l_{U,V\setminus U}(iv)J^l_{V\setminus U,V\setminus U}(iv)^{-1}h^l_{V\setminus U}(iv)\Big)^T y_U \\
&-\frac{1}{2}y_U^T\Big(J^l_{U,U}(iv)-J^l_{U,V\setminus U}(iv)J^l_{V\setminus U,V\setminus U}(iv)^{-1}J^l_{V\setminus U,U}(iv)\Big)y_U\Big\} \\
&\times\int_{y_{V\setminus U}=-\infty}^{y_{V\setminus U}=+\infty}\exp\Big\{-\frac{1}{2}\Big(y_{V\setminus U}+J^l_{V\setminus U,V\setminus U}(iv)^{-1}J^l_{V\setminus U,U}(iv)y_U \\
&-J^l_{V\setminus U,V\setminus U}(iv)^{-1}h^l_{V\setminus U}(iv)\Big)^T J^l_{V\setminus U,V\setminus U}(iv)\Big(y_{V\setminus U} \\
&+J^l_{V\setminus U,V\setminus U}(iv)^{-1}J^l_{V\setminus U,U}(iv)y_U-J^l_{V\setminus U,V\setminus U}(iv)^{-1}h^l_{V\setminus U}(iv)\Big)\Big\}dy_{V\setminus U}
\end{aligned}
\tag{4.15}
$$

But since:

$$
\int_{y_A=-\infty}^{y_A=+\infty}\exp\Big\{-\frac{1}{2}\big(y_A-\mu_A\big)^T J\big(y_A-\mu_A\big)\Big\}dy_A=(2\Pi)^{|\Gamma_A|/2}(\det J)^{-1/2}
$$

Equation 4.15 gives:

$$
\begin{aligned}
&\int_{y_{V\setminus U}=-\infty}^{y_{V\setminus U}=+\infty} \phi(i_V, y_{V\setminus U}, y_U)\, dy_{V\setminus U} \\
&= \sum_{l=1}^{L} (2\Pi)^{|\Gamma_{V\setminus U}|/2} \Big(\det J^l_{V\setminus U, V\setminus U}(i_V)\Big)^{-1/2} \chi_l(i_V) \exp\Big\{ g_l(i_V) \\
&\quad + \frac{1}{2} h^l_{V\setminus U}(i_V)^T J^l_{V\setminus U, V\setminus U}(i_V)^{-1} h^l_{V\setminus U}(i_V) \\
&\quad + \Big(h^l_U(i_V) - J^l_{U, V\setminus U}(i_V) J^l_{V\setminus U, V\setminus U}(i_V)^{-1} h^l_{V\setminus U}(i_V)\Big)^T y_U \\
&\quad - \frac{1}{2} y_U^T \Big(J^l_{U,U}(i_V) - J^l_{U, V\setminus U}(i_V) J^l_{V\setminus U, V\setminus U}(i_V)^{-1} J^l_{V\setminus U, U}(i_V)\Big) y_U \Big\} \\
&= \sum_{l=1}^{L} \chi_l(i_V) \exp\Big\{ \Big( g_l(i_V) + \frac{1}{2}\Big( |\, \Gamma_{V\setminus U}\, | \log(2\Pi) - \log\det J^l_{V\setminus U, V\setminus U}(i_V) \\
&\quad + h^l_{V\setminus U}(i_V)^T J^l_{V\setminus U, V\setminus U}(i_V)^{-1} h^l_{V\setminus U}(i_V)\Big)\Big) \\
&\quad + \Big(h^l_U(i_V) - J^l_{U, V\setminus U}(i_V) J^l_{V\setminus U, V\setminus U}(i_V)^{-1} h^l_{V\setminus U}(i_V)\Big)^T y_U \\
&\quad - \frac{1}{2} y_U^T \Big(J^l_{U,U}(i_V) - J^l_{U, V\setminus U}(i_V) J^l_{V\setminus U, V\setminus U}(i_V)^{-1} J^l_{V\setminus U, U}(i_V)\Big) y_U \Big\} \\
&= \sum_{l=1}^{L} \chi_l(i_V) \exp\Big\{ g'_l(i_V) + h'_l(i_V)^T y_U - y_U^T J'_l(i_V) y_U \Big\}
\end{aligned}
$$

where $g'_l$, $h'_l$ and $J'_l$ for $l = 1, 2, \ldots, L$ are given by:

$$
\begin{aligned}
g'_l(i_V) &= g_l(i_V) + \frac{1}{2}\Big\{ |\, \Gamma_{V\setminus U}\, | \log(2\Pi) - \log\det J^l_{V\setminus U, V\setminus U}(i_V) \\
&\quad + h^l_{V\setminus U}(i_V)^T J^l_{V\setminus U, V\setminus U}(i_V)^{-1} h^l_{V\setminus U}(i_V)\Big\} \\
h'_l(i_V) &= h^l_U(i_V) - J^l_{U, V\setminus U}(i_V) J^l_{V\setminus U, V\setminus U}(i_V)^{-1} h^l_{V\setminus U}(i_V) \\
J'_l(i_V) &= J^l_{U,U}(i_V) - J^l_{U, V\setminus U}(i_V) J^l_{V\setminus U, V\setminus U}(i_V)^{-1} J^l_{V\setminus U, U}(i_V)
\end{aligned}
\tag{4.16}
$$

Thus the marginalisation of $\phi(x_V)$ with respect to $Y_{V\setminus U}$ is a CGM-potential also.

□

## 4.7 Closure of CGM-Potentials

In this section we will show that CGM-potentials are closed under the processes of initialisation and propagation. This being the case we will be able to define the

basic operations we considered in Section 4.6 such that they take CGM-potentials, and only CGM-potentials, as inputs. Closure will also ensure that they only output CGM-potentials. This knowledge will simplify the set of symbolic functions required by the system and ensure the validity and generality of the functions we define.

### 4.7.1 Initialising the System

Associated with each clique $C \in \mathcal{C}$ is a potential function $a_C(x_C)$ which is a function defined on the variables in that clique. Similarly, associated with each separator $S \in \mathcal{S}$ we have a potential function $b_S(x_S)$, which is a function defined on the variables in that separator. We initially let $a_C(x_C) \equiv 1$, for all $C \in \mathcal{C}$ and $b_S(x_S) \equiv 1$ for all $S \in \mathcal{S}$.

For each variable $X_a \in X$, $a = 1, 2, \ldots, k$, we assign $X_a$ to a clique $C$ in the junction tree. We may only assign $X_a$ to a clique which contains all the variables necessary to define the conditional distribution of $X_a$ given its parents - namely both $X_a$ and $X_{pa(a)}$. If more than one suitable clique exists then the choice of which one of them $X_a$ should be assigned to is arbitrary. When a variable $X_a$ is assigned to a clique $C$ the potential function, $a_C(x_C)$, on that clique is updated by multiplying it by the conditional probability density function of $X_a \mid X_{pa(a)}$. When all the variables have been assigned the joint probability density function $f$ of $X$ is as follows:

$$f_K(x) = \prod_{C \in \mathcal{C}} a_C(x_C) \tag{4.17}$$

since $f$ factorises into the conditional distributions of all the variables given their parents. We may trivially express Equation 4.17 as:

$$f_K(x) = \frac{\prod_{C \in \mathcal{C}} a_C(x_C)}{\prod_{S \in \mathcal{S}} b_S(x_S)} \tag{4.18}$$

since $b_S \equiv 1$. We will term Equation 4.18 the *joint system belief.*

**Theorem 34** *Consider a mixed graphical association model in which no continuous variable has a discrete offspring and the conditional probability distributions of the discrete and continuous variables are as given in Section 4.5. Then the potentials following initialisation are CG-potentials and hence CGM-potentials.*

**Proof.** The conditional probability distribution of each discrete variable $X_a$, for $a \in \Delta$, may be expressed as a CG-potential with canonical characteristics $(g(i_{a,pa(a)}), 0, 0)$. Similarly, the conditional probability distribution of each continuous variable $X_a$, for $a \in \Gamma$, may be expressed as a CG-potential with canonical characteristics $(g(i_{pa(a)\cap\Delta}), h(i_{pa(a)\cap\Delta}), J(i_{pa(a)\cap\Delta}))$. The initial clique potentials are thus formed as the product of a series of CG-potentials. By Theorem 17 the product of any two CG-potentials is a CG-potential also. Hence the initial clique potentials are CG-potentials.

The initial separator potentials are defined to be one which may be written as a CG-potential with canonical characteristics $(0, 0, 0)$ and indicator function one. The initial potentials are thus CG-potentials and hence CGM-potentials.

□

Note we could define the conditional distributions of the continuous variables as weighted mixtures of conditional Gaussian distributions. Then, by application of Theorem 30 we could show that the initial potentials are CGM-potentials.

### 4.7.2 Propagating a Schedule

We define the process of propagation as in Section 2.12 and thus it will not be necessary to repeat the proof as to the validity of the propagation method. We will, however, still need to show that the particular functional form of our potential functions is preserved under the application of our propagation algorithm.

**Theorem 35** *Suppose that the initial potentials of a junction tree $\mathcal{T}$ are CGM-potentials then the potentials at any stage of the passage of a full schedule of active flows through $\mathcal{T}$ are CGM-potentials also.*

**Proof.** Recall that the passage of a flow from a clique $C_1$ to an adjacent clique $C_2$ via a separator $S_0$ replaces an original charge $\mathcal{K} = (\{a_C : C \in \mathcal{C}\}, \{b_S : S \in \mathcal{S}\})$ by a new charge $\mathcal{K}^* = (\{a^*_C : C \in \mathcal{C}\}, \{b^*_S : S \in \mathcal{S}\})$, where:

$$
\begin{aligned}
b^*_{S_0} &= \sum_{C_1 \setminus S_0} a_{C_1} \\
a^*_{C_2} &= a_{C_2} \times \lambda_{S_0} \\
\lambda_{S_0} &= \begin{cases} b^*_{S_0}(x_{S_0}) / b_{S_0}(x_{S_0}) & \text{if } b_{S_0}(x_{S_0}) > 0 \\ 0 & \text{if } b_{S_0}(x_{S_0}) = 0 \end{cases}
\end{aligned}
$$

and all other potentials are unaltered. We showed in Theorems 32 and 33 that CGM-potentials are closed under marginalisation and in Theorem 30 that CGM-potentials are closed under multiplication. Thus if $a_{C_1}$, $a_{C_2}$ and $\lambda_{S_0}$ are CGM-potentials then $a^*_{C_2}$ and $b^*_{S_0}$ will be CGM-potentials also. Since the initial potentials of $\mathcal{T}$ are CGM-potentials then we need only show that every $\lambda_{S_0}$ in the passage of a full schedule of active flows is a CGM-potential for the result to follow.

Let us consider two arbitrary adjacent cliques $C_1$ and $C_2$ joined by a separator $S_0$. In the collect evidence phase of a palindromic propagation schedule suppose that the $i-$th active flow is passed from $C_1$ to $C_2$ via $S_0$, and in the distribute evidence phase the $(2k-2-i)-$th active flow is passed from $C_2$ to $C_1$ via $S_0$.

Immediately prior to the $i-$th active flow let $a_{C_1}$, $a_{C_2}$ and $b_{S_0}$ be the potentials on $C_1$, $C_2$ and $S_0$. Suppose that all three potentials are CGM-potentials. During the passage of the $i-$th active flow the ratio $\lambda_{S_0} = b^*_{S_0}/b_{S_0}$ will be calculated. Since $b_{S_0} \equiv 1$ we find, by Theorem 31, that $\lambda_{S_0} = b^*_{S_0}$ is a CGM-potential. Immediately after the passage of the $i-$th active flow the potentials will be $a^*_{C_1} = a_{C_1}$, $a^*_{C_2} = a_{C_2} \times b^*_{S_0}$ and $b^*_{S_0} = \sum_{C_1 \setminus S_0} a_{C_1}$. Since $b^*_{S_0}$ is a CGM-potential $a^*_{C_1}$ and $a^*_{C_2}$ are also CGM-potentials.

Suppose that $i = 1$ then since all the potentials in $\mathcal{T}$ are CGM-potentials the potentials after the passage of the active flow will remain CGM-potentials. Thus, by induction for $i = 2, 3, \ldots, k-1$, every potential in $\mathcal{T}$ will remain a CGM-potential during the collect evidence phase of the propagation schedule.

Now consider the $(2k-2-i)-$th active flow which will take place during the distribute evidence phase of the propagation schedule. Immediately prior to the $(2k-2-i)-$th active flow the potentials on $C_1$ and $S_0$ will be $a^*_{C_1}$ and $b^*_{S_0}$ respectively. Let $\mathcal{T}^+$ be the head of $S_0$ containing $C_2$. Since $C_2$ may have received active flows from other cliques in $\mathcal{T}^+$ then the potential, $a^\dagger_{C_2}$, on $C_2$ immediately prior to the $(2k-2-i)-$th active flow may be written $a^\dagger_{C_2} = a^*_{C_2} \times \phi_{C_2}$ where $\phi_{C_2}$ is the product of update factors passed to $C_2$ by neighbouring cliques in $\mathcal{T}^+$ during active flows $(i+1), (i+2), \ldots, (2k-3-i)$. We know that $a^*_{C_1}$, $a^*_{C_2}$ and $b^*_{S_0}$ are CGM-potentials as are the update factors formed during active flows $(i+1), (i+2), \ldots, (k-1)$. Let us suppose that the update factors formed during active flows $k, (k+1), \ldots, (2k-3-i)$ are CGM-potentials then $a^\dagger_{C_2}$ is a CGM-potential too.

Let $a^{\dagger}_{C_1}$ and $b^{\dagger}_{S_0}$ be the potentials on $C_1$ and $S_0$ immediately after the $(2k-2-i)$−th active flow then:

$$\begin{aligned} b^{\dagger}_{S_0} &= \sum_{C_2 \setminus S_0} a^{\dagger}_{C_2} \\ &= \sum_{C_2 \setminus S_0} a^{*}_{C_2} \times \phi_{C_2} \\ &= \sum_{C_2 \setminus S_0} a_{C_2} \times b^{*}_{S_0} \times \phi_{C_2} \\ &= b^{*}_{S_0} \sum_{C_2 \setminus S_0} a_{C_2} \times \phi_{C_2} \end{aligned}$$

Now since $a^{\dagger}_{C_2}$ is a CGM-potential and CGM-potentials are closed under marginalisation $b^{\dagger}_{S_0}$ is a CGM-potential also. Let $\lambda^{\dagger}_{S_0}$ be the update factor formed during the $(2k-2-i)$−th active flow then by Theorem 31:

$$\begin{aligned} \lambda^{\dagger}_{S_0} &= b^{\dagger}_{S_0} / b^{*}_{S_0} \\ &= \sum_{C_2 \setminus S_0} a_{C_2} \times \phi_{C_2} \end{aligned}$$

but since $a_{C_2}$ and $\phi_{C_2}$ are CGM-potentials and CGM-potentials are closed under multiplication and marginalisation $\lambda^{\dagger}_{S_0}$ is a CGM-potential. Finally since $a^{\dagger}_{C_1} = a^{*}_{C_1} \times \lambda^{\dagger}_{S_0}$ and $a^{*}_{C_1}$ and $\lambda^{\dagger}_{S_0}$ are both CGM-potentials $a^{\dagger}_{C_1}$ is a CGM-potential.

Suppose that $i = k$ then since all the potentials in $\mathcal{T}$ are CGM-potentials the update factor formed during the passage of the $k$−th flow is a CGM-potential and the potentials after the passage of the active flow will thus remain CGM-potentials. Hence, by induction for $i = (k+1), (k+2), \ldots, (2k-2)$, every update factor formed during the passage of the $i$−th active flow will be a CGM-potential and thus the potentials after the passage of the $i$−th active flow will remain CGM-potentials. Hence every potential in $\mathcal{T}$ will remain a CGM-potential during the distribute evidence phase of the propagation schedule.

□

## 4.8 Requirements for the Definition of Symbolic Operations

We showed in Section 4.7 that our system of CGM-potentials is closed under the operations required by our propagation algorithm. We must now consider a

suitable method which will allow us to represent and manipulate CGM-potentials using symbolic algebra. Our chosen language with which we will accomplish this is *Mathematica*.

*Mathematica* is a general computer software system and language intended for mathematical and other applications. Mathematical computations can be divided into three main classes: numerical, symbolic and graphical. *Mathematica* handles these three classes in a unified way. The last two classes are of particular interest to us since they allow the manipulation, symbolic integration and plotting of algebraic formulae. These features will enable us to model probability density functions directly through their formulae.

In Section 4.6 we explored the range of basic operations required to build a PES comprising CGM-potentials. In order to implement these operations as *Mathematica* functions two ingredients are required. First we must decide upon the way in which we will represent a CGM-potential in *Mathematica*. We will term this representation the potential's *symbolic form*. Second we will require a simplification routine to ensure that the potentials generated by, or within, any function are both as simple as possible and match the chosen symbolic form. Given these provisions we may define functions for the basic operations safe in the knowledge that we control both the form of the input potential(s) to a function and the form of the output potential. We can thus construct our functions to prevent the symbolic expressions they output from either "blowing up" or becoming unnecessarily unwieldy. In addition we may cater for all possible situations since CGM-potentials are closed under propagation.

### 4.8.1 Symbolic Form of CGM-Potentials

Recall from Section 4.4 that a CGM-potential $f_A(x_A)$ may be written:

$$\begin{aligned} f_A(x_A) &= f_A(i_A, y_A) \\ &= \sum_{l=1}^{L} \chi_l(i_A) p_l(i_A) \{\det \Sigma_l(i_A)\}^{-1/2} (2\Pi)^{-1/2|\Gamma_A|} \\ &\times \exp\left\{-\frac{1}{2}(y_A - \xi_l(i_A))^T \Sigma_l(i_A)^{-1} (y_A - \xi_l(i_A))\right\} \end{aligned}$$

This potential may be rewritten as:

$$f_A(x_A) = \sum_{l=1}^{L} \chi_l(i_A) \operatorname{sqrt}(r_l(i_A)) \exp\{z_l(x_A)\}$$

where:

$$\begin{aligned} r_l(i_A) &= \frac{(p_l(i_A))^2}{(2\Pi)^{|\Gamma_A|}\det \Sigma_l(i_A)} \\ z_l(x_A) &= -\frac{1}{2}(y_A - \xi_l(i_A))^T \Sigma_l(i_A)^{-1}(y_A - \xi_l(i_A)) \end{aligned} \tag{4.19}$$

Consider a CGM-potential $f_A(i_A^*, y_A)$ for some level $i_A^*$ of $I_A$ for which $f_A$ is defined to be non-zero. In other words no level $i_a^*$, for $a \in A \cap \Delta$, is one for which we have evidence that $f_A$ is zero. We may then drop the indicator functions $\chi_l$ from $f_A$ since they become superfluous. The CGM-potential may thus be expressed as the sum of the product of two functions where one function is the square root of a positive real number and the other function is the exponential of a polynomial in the variables $y_A$ where the maximum exponent of any $y_a$, for $a \in A$, in this polynomial is two. This will be our chosen *symbolic form* for a CGM-potential. Note that it is equivalent to the sum of a series of CG-potentials.

### 4.8.2 Pattern Matching Techniques

*Mathematica* uses patterns to represent classes of expressions (Wolfram, 1991). The basic object that appears in almost all *Mathematica* patterns is _ (pronounced "blank"). The fundamental rule is that _ stands for any expression. The pattern `foo[_]` therefore stands for any expression of the form `foo[`*`anything`*`]`. We may name expressions matching a pattern or parts of a pattern. `x_` is any expression to be named `x`. The pattern `foo[x_]` is a function, `foo`, of an expression to be named `x`. `x:`*`pattern`* is an expression to be named `x` which matches *`pattern`*. The functions `Sqrt` and `Exp` are defined in *Mathematica* as the square root and exponential functions, respectively. We avoid using these functions in our symbolic form since *Mathematica* will evaluate them automatically and our symbolic form may thus be lost. Instead we employ `sqrt` and `exp` as our square root and exponential functions (note that *Mathematica* is case-sensitive). These will not be evaluated by *Mathematica*, however their arguments will be, and the cancelling of functions may occur. For example `sqrt[x] / sqrt[x]` will be automatically replaced by `1`, `sqrt[x] + sqrt[x]` by `2 * sqrt[x]`, `sqrt[5 - 3]` by `sqrt[2]` but `sqrt[4]` will not be replaced by `2`.

We will thus define a CG-potential to be a pattern `cg` where:

```
cg = cg : sqrt[_] * exp[_];
```

Note that we only need define cg as the product of the two functions sqrt and exp. We do not need to explicitly define sqrt[_] as a function of a real number or exp[_] as a function of a polynomial with terms of maximum exponent 2. This is because the closure of CGM-potentials under the operators employed in propagation ensures this to be so. If we want to operate on the individual parts of a CG-potential then we might instead use a pattern:

$$\texttt{cg1} = \texttt{cg1} : \texttt{sqrt[x1\_]} * \texttt{exp[y1\_]};$$

The CG-potential could then be accessed using the expression cg1, and the arguments of the sqrt and exp functions as x1 and y1 respectively.

The function MatchQ[*expr*, *form*] enables us to test to see if an expression *expr* matches a pattern *form* returning True or False as appropriate. We may define a function iscgQ to check to see if an expression is a CG-potential or not as follows:

$$\texttt{iscgQ[x\_]} := \texttt{MatchQ[x, cg]};$$

iscgQ[sqrt[x]*exp[y]] will thus return True while iscgQ[x] will return False. A function foo[cg] := 3, say, defined with argument cg will only be evaluated on expressions of the form cg. Expressions which are not of the form cg will be returned unaltered. Thus foo[sqrt[x]*exp[y]] would return 3 while foo[x] would simply return foo[x].

These basic pattern matching capabilities will enable us to write functions which operate on CG-potentials safe in the knowledge that what we pick up as a CG-potential really *is* a CG-potential. A CGM-potential is a sum of CG-potentials. We may thus extend these techniques in an obvious way to apply to any CGM-potential of a particular number of terms. We may not, however, define a pattern to match a CGM-potential with an arbitrary number of terms as simply as we defined cg, or extract each of the arguments of its sqrt or exp functions as simply as we did with cg1. This will not pose a problem, however, as we may apply a slightly different pattern matching technique to obtain the same result. Since we already know how to deal with CG-potentials let us consider how we may deal with CGM-potentials which are not CG-potentials also. In order to test to see if an expression is a CGM-potential of two or more terms we must test to see if it is a summation and if every term in that summation is a CG-potential. The function iscgmQ will perform this test:

```
iscgmQ[x_]  :=  ( Head[x] === Plus ) &&
                ( Apply[And, Map[iscgQ, Apply[List, x]]] );
```

In order to define a function which may only be applied to a CGM-potential of two or more terms we may apply a rule to its argument using the \; construct. Thus a function `foo` defined as `foo[x_\;iscgmQ[x]] := 3` applies the rule `iscgmQ` to the input argument `x` and returns the value `3` if `x` is a CGM-potential of two or more terms and `foo[x]` otherwise. The function `exppart` may be applied to CG-potentials only:

```
exppart[sqrt[x_] * exp[y_]] := y
```

It returns the argument of the `exp` part of the CG-potential. If we change the summation operator of a CGM-potential to a list operator we may then apply the `exppart` function to every element of that list and hence determine a list of the arguments of the `exp` functions. The code `Map[exppart, Apply[List, x]]` performs this on a CGM-potential `x`. A list of the arguments of the `sqrt` functions may be determined in a similar way.

## 4.9 Symbolic Operations

Having discussed our requirements for the creation of symbolic operators in the last section we will now outline their construction in *Mathematica*.

### 4.9.1 Initial Representation of Conditional Distributions

Recall from Section 4.5 that discrete random variables are forbidden from having continuous parents and they are defined, by conditioning on their discrete parents, as multinomial distributions. Consider a random variable $I_a$, for $a \in \Delta$, with parents $X_{pa(a)} = I_{pa(a)}$ and distribution, as given in Equation 4.6, thus:

$$f_{a|pa(a)}\left(i_a; i_{pa(a)}\right) = p\left(i_a; i_{pa(a)}\right)$$

Then we may write the probability density function of $I_a \mid X_{pa(a)}$ as:

$$\sqrt{p(i_a; i_{pa(a)})^2} \exp\{0\}$$

in order that it will match our chosen symbolic form. The *Mathematica* function `cons` will generate this symbolic form:

```
cons[p_] := sqrt[p ^2] * exp[0];
```

The initial distribution of each continuous random variable is conditionally Normal given its discrete and continuous parents. Consider a random variable $Y_a$, for $a \in \Gamma$, with parents $X_{pa(a)} = (I_{pa(a)}, Y_{pa(a)})$ and distribution, as given in Equation 4.7, thus:

$$Y_a \mid X_{pa(a)} \sim N\left(\alpha(i_{pa(a)}) + \beta(i_{pa(a)})^T y_{pa(a)}, \gamma(i_{pa(a)})\right)$$

Then we may write the probability density function of $Y_a \mid X_{pa(a)}$ as:

$$\sqrt{\frac{1}{2\Pi\gamma(i_{pa(a)})}} \exp\left\{\frac{-1}{2\gamma(i_{pa(a)})}\left(y_a - \alpha(i_{pa(a)}) - \beta(i_{pa(a)})^T y_{pa(a)}\right)^2\right\}$$

which matches our choice of symbolic form for a CG-potential. We will find it most convenient to expand all the terms within the square root and exponential operators wherever possible. This will allow any required cancelling to occur. The following *Mathematica* function defines a univariate normal distribution in the desired symbolic form.

```
uninorm[x_, mu_, sigsq_] := Block[{const, xmu},
      const = sqrt[1 / (2 * Pi * sigsq)];
      xmu = ExpandAll[(-1/ (2*sigsq))*((x-mu)^2)];
      Return[const * exp[xmu]];
]
```

In order that we may maximise the accuracy of our functions and prevent roundoff errors from interfering with their proper functioning we define all numeric values in *Mathematica* as either integers or rational numbers. In addition mathematical constants, such as pi, and functions, such as exponentials and square roots, are kept in symbolic form to prevent the generation of irrational numbers or roundoff errors. The initial conditional distributions of $B$ and $C$ in the waste incinerator problem are thus defined as follows:

```
bdist =  {{{},{b}} , {{cons[17/20]}, {cons[3/20]}}};
cdist =  {{{c},{b}}, {{uninorm[c, -2, 1/10]},
                      {uninorm[c, -1, 3/10]}}};
```

These objects, on call of `cons` and `uninorm`, then become:

```
bdist = {{{},{b}} , {{exp[0]*sqrt[289 / 400]},
                     {exp[0]*sqrt[9 / 400]}}};
cdist = {{{c},{b}}, {{exp[-20 - 20*c - 5*c^2]*sqrt[5/Pi]},
                     {exp[-5/3 - (10*c)/3 - (5*c^2)/3]*
                      sqrt[5/(3*Pi)] }}};
```

### 4.9.2 Multiplication

The *Mathematica* implementation of the multiplication operator may be defined in a similar way to that used in the discrete exact case. Assuming that the potentials have been extended and rearranged such that their organisation matches, we may assign the list of variables from the first function to the first element of the output potential and the product of the two potential tables to the second element of the output potential. In order that the resulting potentials are in as simple a form as possible, and that that form matches our symbolic form, we additionally apply a simplification operator `mcgm` to every cell in the output potential table. The multiplication operator, `multdists`, may be defined as:

```
multdists[dist1_, dist2_] := Block[{vars, data, len},
      vars = dist1[[1]];
      len = Length[vars[[2]]];
      data = Map[mcgm, dist1[[2]]*dist2[[2]],{len}];
      Return[Join[{vars}, {data}]];
]
```

The simplification operator, `mcgm`, needs to be able to handle every possible combination of CGM-potentials. The most elegant way to achieve this is to employ the use of pattern matching and define `mcgm` as a function which will recursively call itself on sub-arguments of its input argument until all possible simplifications have been carried out. We define six rules, M1-M6, to achieve this goal. Rule M1 is simply used to preserve the list structure of *Mathematica*, rules M2-M6 are simplification operators. Rules M2-M5 deal with the product of two CG-potentials - the simplest structure in our symbolic form. Rule M2 simplifies the product of two CG-potentials with different arguments to the functions `exp` and `sqrt`. Since *Mathematica* will only match an expression to a pattern if the expression is in exactly the same form as the pattern rules M3-M5 are additionally required.

These deal with the cases where either one or both of the arguments of the `exp` and `sqrt` functions are identical in the two input CG-potentials. In this case *Mathematica* will automatically replace the product of the two identical functions with the square of one of them and a different pattern is thus created. The final rule, M6, provides the recursion. It deals with the product of two potentials where at least one of them is a CGM-potential (the other may be either a CG or CGM-potential). This rule will repeatedly be applied to expand the input expression into a summation of terms where each term consists of a function call to `mcgm` with an argument which corresponds to the product of two CG-potentials. On the call of `mcgm` to each of these terms a CGM-potential requiring no further simplification will be output.

```
mcgm[{x_}] := {mcgm[x]};                                          (M1)
mcgm[sqrt[x1_]*exp[y1_]*sqrt[x2_]*exp[y2_]] :=
     sqrt[x1*x2] * exp[y1+y2];                                    (M2)
mcgm[sqrt[x1_]*sqrt[x2_]*(exp[y_]^2)] :=
     sqrt[x1*x2] * exp[2 * y];                                    (M3)
mcgm[(sqrt[x_]^2)*exp[y1_]*exp[y2_]] :=
     sqrt[x^2] * exp[y1+y2];                                      (M4)
mcgm[(sqrt[x_]^2)*(exp[y_]^2)] :=
     sqrt[x^2] * exp[2 * y];                                      (M5)
mcgm[x_ * (y_ + z_)] := mcgm[x*y] + mcgm[x*z];                    (M6)
```

It should be noted that while an expression `sqrt[x]^2` will not be matched by a pattern `sqrt[x_]*sqrt[y_]` even though that expression may be rewritten `sqrt[x]*sqrt[x]` which then matches the pattern, rules of associativity and commutativity *are* used when pattern matching. The expressions `sqrt[x]*exp[y]` and `exp[y]*sqrt[x]` will therefore both be matched correctly by the pattern `exp[x_]*sqrt[y_]`.

### 4.9.3 Division

The *Mathematica* function to perform the division of two CGM-potentials may be defined in a way which is similar to the multiplication operator. We additionally add a line to simplify the case where the numerator and denominator are numerically equivalent, replace the multiplication sign with a division sign and employ the use of a different simplification function. The division operator `divdists`,

which divides two CGM-potentials which have first been extended and rearranged to match, is thus as follows:

```
divdists[dist1_, dist2_] := Block[{vars, data, len, temp},
      vars = dist1[[1]];
      len = Length[vars[[2]]];
      temp = dist1[[2]];
      If[N[temp == dist2[[2]]], (temp = dist2[[2]])];
      data = Map[dcgm, temp / dist2[[2]], {len}];
      Return[Join[{vars}, {data}]];
]
```

We shall consider the definition of the simplification function, dcgm, in two stages. Firstly the simpler case in which a CG or CGM-potential is divided by a CG-potential, and secondly the more complex situation where we require the ratio of two CGM-potentials. Again we also have a rule, D1, to preserve the list structure of *Mathematica*.

```
dcgm[{x_}] := {dcgm[x]};                                    (D1)
```

Rules D2-D5 deal with the cases where one CG-potential is being divided by another CG-potential. Rule D2 corresponds to the case where both numerator and denominator are equal and automatically cancel. Rules D3 and D4 deal with the cases where either the exponential functions or the square root functions in both the numerator and denominator equate and cancel. Rule D5 corresponds to the situation where the numerator and denominator are CG-potentials with different exponential and square root terms so no automatic cancellation occurs. Rules D6-D8 are recursive and are used to divide a CGM-potential by a CG-potential. In rule D6 no cancellation has occurred, in rule D7 the exponential operators in numerator and denominator have cancelled, and in rule D8 the square root operators in numerator and denominator have cancelled. Each rule calculates the first term and reapplies dcgm to simplify the remaining terms.

```
dcgm[1] := sqrt[1]*exp[0];                                  (D2)
dcgm[sqrt[x1_] / sqrt[x2_]] := sqrt[x1/x2]*exp[0];           (D3)
dcgm[exp[y1_] / exp[y2_]] := sqrt[1]*exp[y1-y2];             (D4)
```

```
dcgm[(sqrt[x1_]*exp[y1_]) / (sqrt[x2_]*exp[y2_])] :=
      sqrt[x1/x2]*exp[y1-y2];                                    (D5)
dcgm[(sqrt[x1_]*exp[y1_] + z_) / (sqrt[x2_]*exp[y2_])] :=
      sqrt[x1/x2]*exp[y1-y2] + dcgm[z/(sqrt[x2]*exp[y2])];       (D6)
dcgm[(sqrt[x1_] + z_) / (sqrt[x2_])] :=
      sqrt[x1/x2]*exp[0] + dcgm[z/(sqrt[x2])];                   (D7)
dcgm[(exp[y1_] + z_) / (exp[y2_])] :=
      sqrt[1]*exp[y1-y2] + dcgm[z/(exp[y2])];                    (D8)
```

In order to calculate the ratio of two CGM-potentials we require four additional functions. The functions `iscgQ` and `iscgmQ` determine if a function is a CG or a CGM-potential respectively, or not. The functions `exppart` and `sqrtpart` obtain the arguments of the exponential and square root operators, respectively.

```
iscgQ[x_] := MatchQ[x, sqrt[_]*exp[_]];
iscgmQ[x_] := (Head[x] === Plus) &&
      (Apply[And, Map[iscgQ, Apply[List, x]]]);
exppart[exp[x_]*sqrt[y_]] := x;
sqrtpart[exp[x_]*sqrt[y_]] := y;
```

Rule D9 calculates the ratio of two CGM-potentials using these four functions. The functions `iscgQ` and `iscgmQ` are used to check that the numerator and denominator are both CGM-potentials. If they are then rule D9 is applied.

```
dcgm[(x_ /; iscgmQ[x]) / (y_ /; iscgmQ[y])] := Block[{xseries,
      yseries, xexp, yexp, xsqrt, ysqrt, zexp, zsqrt, zlist},
      xseries = Apply[List, x];
      yseries = Apply[List, y];
      xexp = Map[exppart, xseries];
      yexp = Map[exppart, yseries];
      xsqrt = Map[sqrtpart, xseries];
      ysqrt = Map[sqrtpart, yseries];
      zexp = Map[exp, Table[xexp, {Length[yexp]}] - yexp, {2}];
      zsqrt =Map[sqrt,Table[xsqrt, {Length[ysqrt]}]/ysqrt,{2}];
      zlist = Apply[Intersection, zexp * zsqrt];
      Return[Apply[Plus, zlist]];
]                                                                (D9)
```

Consider the following example which explains the logic behind rule D9. Let $a$, $b$, $c$, and $d$ be four CG-potentials. The product of the CGM-potentials $(a+b)$ and $(c+d)$ is $(ac+ad+bc+bd)$. Now suppose we wish to determine the ratio of this product and $(a+b)$ how might we, in general, proceed? One possible way might be to represent the numerator as a list $\{ac, ad, bc, bd\}$ and the denominator as a list $\{a, b\}$. We can then form a table in which every column in the table corresponds to an element in the numerator, and each row in the table corresponds to an element in the denominator. Every cell in the table is defined as the ratio of the appropriate elements of the numerator and denominator defined by the row/column combination. For our example we obtain the table $\{\{c, d, bc/a, bd/a\}, \{ac/b, ad/b, c, d\}\}$. In order to find the CGM-potential resulting from the division we find the intersection of the rows $\{c, d\}$ and convert the list back into a summation $(c+d)$. Rule D9 carries out this logic. We convert the CGM-potentials into lists of CG-potentials (`xseries` and `yseries`). Split the lists of CG-potentials into lists of arguments to the exponential and square root functions (`xexp`, `yexp`, `xsqrt`, and `ysqrt`). Form tables of the ratios of the exponential and square root functions in the numerators and denominators (`zexp` and `zsqrt`). Finally we form the product of these two tables and find the intersection between rows of the tables. The resulting list (`zlist`) is then converted to a CGM-potential and output.

In case the `exp` or `sqrt` functions in both numerator and denominator cancel we will require two further rules, D10 and D11, to cover these possibilities. The functions `isexpQ`, `isexpmQ`, `issqrtQ`, and `issqrtmQ` are used to match these cases. If both the numerator and denominator are CGM-potentials which equate fully then *Mathematica* will cancel them and automatically return the value 1. Rule D2 will then be applied.

```
dcgm[(x_ /; isexpmQ[x]) / (y_ /; isexpmQ[y])] := Block[{
        xseries, yseries, xexp, yexp, zexp, zlist},
        xseries = Apply[List, x];
        yseries = Apply[List, y];
        xexp = Map[exppart, xseries];
        yexp = Map[exppart, yseries];
        zexp = Map[exp, Table[xexp, {Length[yexp]}] - yexp, {2}];
        zlist = sqrt[1] * Apply[Intersection, zexp];
        Return[Apply[Plus, zlist]];
]                                                          (D10)
```

```
dcgm[(x_ /; issqrtmQ[x]) / (y_ /; issqrtmQ[y])] := Block[{
      xseries, yseries, xsqrt, ysqrt, zsqrt, zlist},
      xseries = Apply[List, x];
      yseries = Apply[List, y];
      xsqrt = Map[sqrtpart, xseries];
      ysqrt = Map[sqrtpart, yseries];
      zsqrt =Map[sqrt,Table[xsqrt, {Length[ysqrt]}]/ysqrt,{2}];
      zlist = exp[0] * Apply[Intersection, zsqrt];
      Return[Apply[Plus, zlist]];
]                                                          (D11)

isexpQ[x_] := MatchQ[x, exp[_]];
isexpmQ[x_] := (Head[x] === Plus) &&
      (Apply[And, Map[isexpQ, Apply[List, x]]]);
issqrtQ[x_] := MatchQ[x, sqrt[_]];
issqrtmQ[x_] := (Head[x] === Plus) &&
      (Apply[And, Map[issqrtQ, Apply[List, x]]]);
```

### 4.9.4 Marginalisation

The final symbolic operator we need consider to form a propagation algorithm is that of marginalisation. Since we are dealing with a mixed PES the marginalisation of a potential function may be with respect to both discrete and continuous random variables. The order with which marginalisation is carried out is, in theory, irrelevant. We do, however, apply the rule that the list of continuous random variables should be marginalised over prior to marginalisation with respect to the discrete random variables. The advantage of this approach is that it keeps the individual functions we need to marginalise with respect to continuous variables as simple as possible. Moreover, for simplicities sake, we marginalise with respect to a single random variable at a time.

The *Mathematica* function `marginalise` carries out the marginalisation. First it determines which variables are to be marginalised with respect to, by comparison with the lists of variables we already have evidence on (`fulldisvars` and `fullctsvars`). The list of continuous variables to be marginalised with respect to is put into the global list `globalctsvars`. The function `margctsvars` is then

applied to marginalise with respect to the continuous variables. If there are no discrete variables to be marginalised with respect to the resulting potential is output. Otherwise the potential is marginalised with respect to the appropriate list of discrete variables, variable by variable. This is achieved by first rearranging the potential table such that a discrete variable to be marginalised with respect to is on the outermost layer of the table. Then the levels of the discrete variable may be summed over. The resulting potential is simplified using the function `scgm` and then output.

```
marginalise[table_, vars_] := Block[{data, disvars, newctsvars,
      newdisvars, newvars, numdisvars, neword},
      data = table[[2]];
      globalctsvars = vars[[1]];
      disvars = comp[vars[[2]], fulldisvars];
      newctsvars = Complement[table[[1, 1]], globalctsvars];
      newdisvars = comp[table[[1, 2]], vars[[2]]];
      newvars = Join[{newctsvars}, {newdisvars}];
      numdisvars = Length[table[[1, 2]]];
      globalctsvars = comp[globalctsvars, fullctsvars];
      If [globalctsvars != {}, data = Map[margctsvars, data,
                                          {numdisvars}]];
      If [disvars == {}, Return[Join[{newvars},{data}]]];
      neword = Join[disvars, newdisvars];
      data = rearrdis[neword, {{newctsvars, table[[1, 2]]},
                               data}][[2]];
      numdisvars = Length[disvars];
      While[numdisvars > 0,
            data = Apply[Plus, data];
            numdisvars = numdisvars - 1
      ];
      data = Map[scgm, data, {Length[disvars] }];
      Return[{newvars, data}];
]
```

The function `margctsvars` called with the potential function `dist` calls the function `marg` with the potential function `dist` and `globalctsvars`, the list of continuous variables to be marginalised with respect to.

```
margctsvars[dist_] := marg[dist, globalctsvars];
```

The function `marg` loops over the list of continuous variables in turn and calls the symbolic integration function `integrate`. The resulting potential is then simplified using `scgm`.

```
marg[dist_, vars_] := Block[{varlist, result, curvar},
        varlist = vars;
        result = dist;
        While[ Length[varlist] > 0,
                curvar = varlist[[1]];
                result = scgm[integrate[result, curvar]];
                varlist = Rest[varlist]
        ];
        Return[result];
]
```

Marginalisation of a potential function with respect to a continuous random variable requires the integration of the potential function with respect to that random variable over its entire space. *Mathematica* possesses its own integration function, but this is too general, and hence slow, for our purposes. In addition it will be unable to interpret our symbolic form correctly. We therefore define our own integration function - `integrate`. Four rules I1-I4 are used to define the integration of a CGM-potential with respect to a continuous random variable.

```
integrate[{a_}, x_] := {integrate[a, x]};                              (I1)
integrate[a_ + b_, x_] := integrate[a, x] + integrate[b, x];           (I2)
integrate[sqrt[i1_]*exp[i2_], x_] :=
```

```
        integ[sqrt[i1]*exp[Collect[i2,x]], x];                    (I3)
integ[sqrt[k_]*exp[a_ + (b_ * x_) + (c_ * x_^2)], x_] :=
        sqrt[-(k*Pi)/c]*exp[ExpandAll[a - ((b^2)/(4*c))]];        (I4)
```

Rule I1 is used to preserve the list structure. Rule I2 calls itself recursively such that the integration of a CGM-potential is calculated as the sum of the integration of its component CG-potentials. Rule I3 prepares for the integration of a CG-potential. It calls the function `integ` with the CG-potential it takes as an input. Additionally it uses the `Collect` function to collect together the terms of the variable `x`, say, to be marginalised with respect to in the argument to the exponential function. This ensures that the argument of the exponential function is represented as a quadratic in `x`. Rule I4 defines `integ` which integrates a CG-potential in this form. It applies the following result which holds for $c < 0$:

$$\begin{aligned}
\int_{x=-\infty}^{x=+\infty} \sqrt{k}\ \mathrm{Exp}\left\{a + bx + cx^2\right\} dx &= \int_{x=-\infty}^{x=+\infty} \sqrt{k}\ \mathrm{Exp}\left\{\left(a - \frac{b^2}{4c}\right) + c\left(x + \frac{b}{2c}\right)^2\right\} dx \\
&= \sqrt{\frac{-k\Pi}{c}}\ \mathrm{Exp}\left\{a - \frac{b^2}{4c}\right\} \\
&\quad \times \int_{x=-\infty}^{x=+\infty} \frac{1}{\sqrt{\frac{-\Pi}{c}}}\ \mathrm{Exp}\left\{c\left(x + \frac{b}{2c}\right)^2\right\} dx \\
&= \sqrt{\frac{-k\Pi}{c}}\ \mathrm{Exp}\left\{a - \frac{b^2}{4c}\right\}
\end{aligned} \tag{4.20}$$

All that is left to define is our simplification function `scgm`. This function is used to simplify CGM-potentials after they have been summed over discrete variables or integrated over continuous ones. It is used to retain the symbolic form of the CGM-potential. Rule S1 preserves the list structure of *Mathematica*. Rule S2 combines two CG-potentials with the same exponential term. Rules S3 and S4 put a numeric multiple of a CG-potential into the correct symbolic form. Rule S3 deals with the case where there are other terms to be simplified, rule S4

where there are not. The final rule is applied if no other rule needs to be fitted and returns its input unchanged.

```
scgm[{x_}] := {scgm[x]};                                              (S1)
scgm[sqrt[x1_]*exp[y_] + sqrt[x2_]*exp[y_] + z__] :=
     scgm[sqrt[x1 + 2*Sqrt[x1*x2] + x2 ]*exp[y] + z];                 (S2)
scgm[(n_ /; NumberQ[n])*sqrt[x_]*exp[y_] + z__] :=
     scgm[sqrt[n^2 * x]*exp[y] + z];                                  (S3)
scgm[(n_ /; NumberQ[n])*sqrt[x_]*exp[y_]] :=
     scgm[sqrt[n^2 * x]*exp[y]];                                      (S4)
scgm[x_] := x;                                                        (S5)
```

## 4.10 Symbolic Implementation of the Waste Incinerator Problem

### 4.10.1 Initialisation

Let us assume that the assignment of variables to cliques is as given in Table 4.6. The initial clique potentials for the waste incinerator problem are given in Table 4.7. These potentials are represented in our chosen symbolic form.

| Variables Assigned | Clique |
|---|---|
| C, B | { { C }, { B } } |
| D | { { D, E }, { B, W } } |
| L | { { D, L }, { } } |
| E, F, W | { { E }, { F, W } } |
| Mi | { { D, Mi }, { W } } |
| M0 | { { D, Mi }, { M0 } } |

Table 4.6: The assignment of variables to cliques in the waste incinerator problem.

| Initial Clique Potential for Clique { { C }, { B } } | |
|---|---|
| B=0 | $\text{sqrt}\left[\frac{289}{80\Pi}\right] * \exp\left[-20 - 20c - 5c^2\right]$ |
| B=1 | $\text{sqrt}\left[\frac{3}{80\Pi}\right] * \exp\left[-\frac{5}{3} - \frac{10c}{3} - \frac{5c^2}{3}\right]$ |

| Initial Clique Potential for Clique { { D, E }, { B, W } } | | |
|---|---|---|
| B=0 | W=0 | $\text{sqrt}\left[\frac{50}{3\Pi}\right] * \exp\left[-\frac{4225}{6} + \frac{650d}{3} - \frac{50d^2}{3} - \frac{650e}{3} + \frac{100de}{3} - \frac{50e^2}{3}\right]$ |
| | W=1 | $\text{sqrt}\left[\frac{25}{2\Pi}\right] * \exp\left[-450 + 150d - \frac{25d^2}{2} - 150e + 25de - \frac{25e^2}{2}\right]$ |
| B=1 | W=0 | $\text{sqrt}\left[\frac{5}{\Pi}\right] * \exp\left[-\frac{1125}{4} + 75d - 5d^2 - 75e + 10de - 5e^2\right]$ |
| | W=1 | $\text{sqrt}\left[\frac{5}{\Pi}\right] * \exp\left[-245 + 70d - 5d^2 - 70e + 10de - 5e^2\right]$ |

| Initial Clique Potential for Clique { { D, L }, { } } |
|---|
| $\text{sqrt}\left[\frac{2}{\Pi}\right] * \exp\left[-18 + 6d - \frac{d^2}{2} + 12l - 2dl - 2l^2\right]$ |

| Initial Clique Potential for Clique { { E }, { F, W } } | | |
|---|---|---|
| F=0 | W=0 | $\text{sqrt}\left[\frac{90250}{49\Pi}\right] * \exp\left[-380250 - 195000e - 25000e^2\right]$ |
| | W=1 | $\text{sqrt}\left[\frac{1128125}{98\Pi}\right] * \exp\left[-256000 - 160000e - 25000e^2\right]$ |
| F=1 | W=0 | $\text{sqrt}\left[\frac{50}{49\Pi}\right] * \exp\left[-800 - 4000e - 5000e^2\right]$ |
| | W=1 | $\text{sqrt}\left[\frac{625}{98\Pi}\right] * \exp\left[-1250 - 5000e - 5000e^2\right]$ |

| Initial Clique Potential for Clique { { D, Mi }, { W } } | |
|---|---|
| W=0 | $\text{sqrt}\left[\frac{50}{\Pi}\right] * \exp\left[-\frac{25}{2} + 50mi - 50mi^2\right]$ |
| W=1 | $\text{sqrt}\left[\frac{100}{\Pi}\right] * \exp\left[-25 - 100mi - 100mi^2\right]$ |

| Initial Clique Potential for Clique { { D, Mi, M0 }, { } } |
|---|
| $\text{sqrt}\left[\frac{250}{\Pi}\right] * \exp\left[-250d^2 - 500dmi - 250mi^2 + 500dm0 + 500mim0 - 250m0^2\right]$ |

Table 4.7: Initial clique potentials for the waste incinerator problem where: $B = \{0,1\} =$ { Stable, Unstable }, $F = \{0,1\} =$ {Intact, Defective}, and $W = \{0,1\} =$ { Industrial, Household }.

### 4.10.2 Propagation

The propagation schedule for the incinerator example is presented in Table 4.8:

| No. | Source Clique | Separator | Sink clique |
|---|---|---|---|
| 1 | { { C }, { B } } | { { }, { B } } | { { D, E }, { B, W } } |
| 2 | { { D, L }, { } } | { { D } , { } } | { { D, E }, { B, W } } |
| 3 | { { E }, { F, W } } | { { E }, { W } } | { { D, E }, { B, W } } |
| 4 | { { D, Mi }, { M0 } } | { { D, Mi }, { } } | { { D, Mi }, { W } } |
| 5 | { { D, Mi }, { W } } | { { D }, { W } } | { { D, E }, { B, W } } |
| 6 | { { D, E }, { B, W } } | { { D }, { W } } | { { D, Mi }, { W } } |
| 7 | { { D, Mi }, { W } } | { { D, Mi }, { } } | { { D, Mi }, { M0 } } |
| 8 | { { D, E }, { B, W } } | { { E }, { W } } | { { E }, { F, W } } |
| 9 | { { D, E }, { B, W } } | { { D }, { } } | { { D, L }, { } } |
| 10 | { { D, E }, { B, W } } | { { }, { B } } | { { C }, { B } } |

Table 4.8: Propagation schedule for the waste incinerator problem.

Following the propagation of a full schedule of flows the potential on each clique (or separator) is the joint probability density function of the variables in that clique (or separator). We may thus form the marginal distribution of any variable by marginalising the potential function of any clique (or separator) that contains it, over the other variables. The marginal distribution of any continuous variable will, in general, be a CGM-distribution. We may thus determine the moments of any continuous marginal distribution by pattern matching. Since *Mathematica* supports graphical capabilities we may also graph the distribution of any variable we wish.

Let $\lambda$ be a normalisation constant for the system (given no evidence $\lambda$ is equal to one). Suppose we wish to find the first two moments of a CGM-distribution which is the marginal distribution of a continuous variable. The moments of the

CGM-distribution may be determined from the moments of the individual CG-distributions of which it is comprised. Suppose that each CG-distribution is in our symbolic form then the mean of each CG-distribution may be determined as follows:

$$\begin{aligned}\int_{x=-\infty}^{x=+\infty} x\sqrt{\frac{k}{\lambda^2}}\,\mathrm{Exp}\left\{a+bx+cx^2\right\}dx &= \sqrt{\frac{-k\Pi}{c\lambda^2}}\,\mathrm{Exp}\left\{a-\frac{b^2}{4c}\right\} \\ &\times \int_{x=-\infty}^{x=+\infty} x\sqrt{\frac{-c}{\Pi}}\,\mathrm{Exp}\left\{c\left(x+\frac{b}{2c}\right)^2\right\}dx \\ &= \frac{-b}{2c}\sqrt{\frac{-k\Pi}{c\lambda^2}}\,\mathrm{Exp}\left\{a-\frac{b^2}{4c}\right\}\end{aligned} \tag{4.21}$$

Similarly the mean square may be determined as:

$$\begin{aligned}\int_{x=-\infty}^{x=+\infty} x^2\sqrt{\frac{k}{\lambda^2}}\,\mathrm{Exp}\left\{a+bx+cx^2\right\}dx &= \sqrt{\frac{-k\Pi}{c\lambda^2}}\,\mathrm{Exp}\left\{a-\frac{b^2}{4c}\right\} \\ &\times \int_{x=-\infty}^{x=+\infty} x^2\sqrt{\frac{-c}{\Pi}}\,\mathrm{Exp}\left\{c\left(x+\frac{b}{2c}\right)^2\right\}dx \\ &= \left(\frac{b^2-2c}{4c^2}\right)\sqrt{\frac{-k\Pi}{c\lambda^2}}\,\mathrm{Exp}\left\{a-\frac{b^2}{4c}\right\}\end{aligned} \tag{4.22}$$

The *Mathematica* function, `mean`, which determines the mean of a CGM-potential using Equation 4.21 is as follows:

```
mean[dist_] := Block[{var, data, mu},
      var = dist[[1, 1, 1]];
      data = dist[[2, 1]];
      mu = meanint[data, var];
      Return[mu];
]
```

It determines the variable and marginal potential from its input. It then calls `meanint` with these arguments. The function `meanint` is defined by two rules. The first is recursive and splits the the CGM-potential into its component CG-potentials. The second rule applies Equation 4.21, effectively carrying out the required integration. In order to determine the mean of a continuous variable's marginal distribution we find the mean of its marginal potential and divide by the normalisation constant $\lambda$.

```
meanint[a_ + b_, x_] := meanint[a, x] + meanint[b, x];
meanint[sqrt[k_]*exp[a_ + (b_ * x_) + (c_ * x_^2)], x_] :=
      -(b/(2*c))*sqrt[(-K*Pi)/c]
      exp[ExpandAll[a - ((b^2)/(4*c))]];
```

The analogous routines, `meansq` and `msq`, to find the mean square of a CG-potential based on Equation 4.22 are given as follows:

```
meansq[dist_] := Block[{var, data, msquare},
      var = dist[[1, 1, 1]];
      data = dist[[2, 1]];
      msquare = msq[data, var];
      Return[msquare];
]
```

```
msq[a_ + b_, x_] := msq[a, x] + msq[b, x];
msq[sqrt[k_]*exp[a_ + (b_ * x_) + (c_ * x_^2)], x_] :=
      sqrt[(-k*Pi)/c]*(((b^2) - (2*c)) / (4 * (c^2)))
      *exp[ExpandAll[a - ((b^2)/(4*c))]];
```

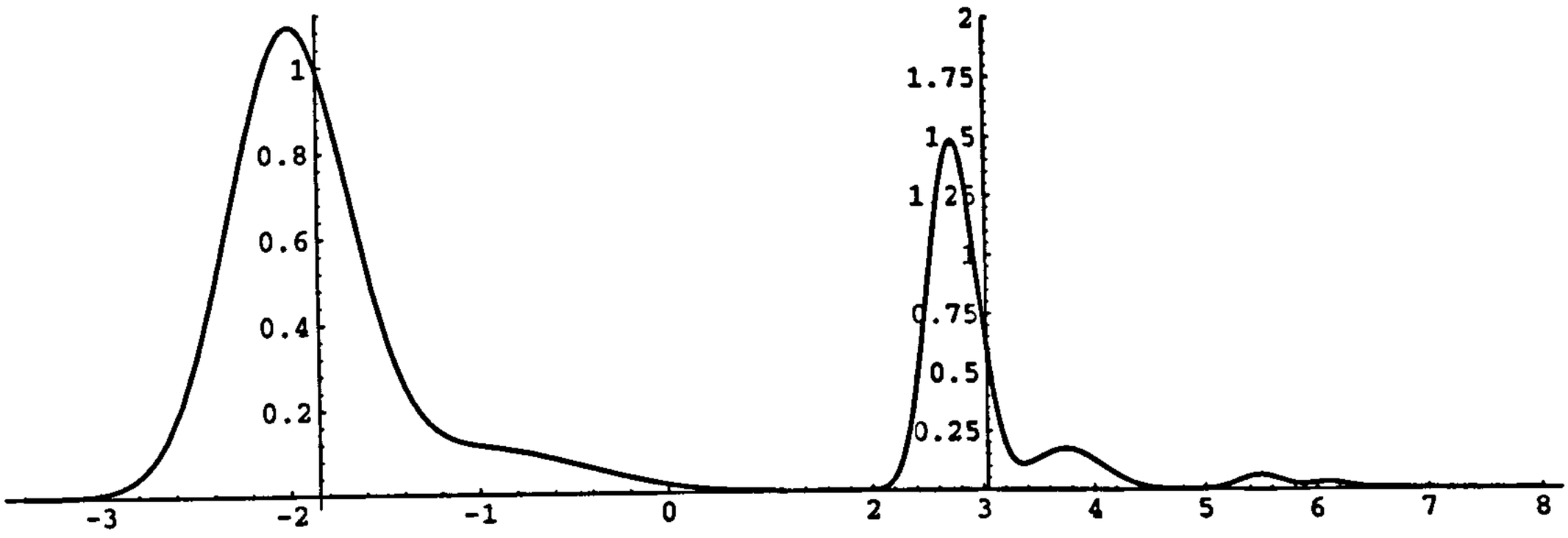
Again, the mean square of a CG-potential should be divided by the normalisation constant to determine the mean square of the corresponding CG-distribution. The variance of the CG-distribution may be determined from its first two moments using the well known formula $\mathrm{Var}(X) = E[X^2] - (E[X])^2$. The means and variances of the continuous variables in the waste incinerator problem, given no evidence, are presented in Table 4.9.

| Variable | Mean | Variance |
|---|---|---|
| C: $CO_2$ in Emission | $\frac{-37}{20} = -1.850000$ | $\frac{103}{400} = 0.257500$ |
| D: Emission of Dust | $\frac{851}{280} = 3.039286$ | $\frac{29052551}{49000000} = 0.592909$ |
| E: Filter Efficiency | $\frac{-911}{280} = -3.253571$ | $\frac{24623051}{49000000} = 0.502511$ |
| L: Light Penetrability | $\frac{829}{560} = 1.480357$ | $\frac{78052551}{196000000} = 0.398227$ |
| Mi: Metal in Waste | $\frac{-3}{14} = -0.214286$ | $\frac{2063}{9800} = 0.210510$ |
| M0: Emission of Metal | $\frac{113}{40} = 2.825000$ | $\frac{5180793}{7000000} = 0.740113$ |

Table 4.9: Means and variances of continuous variables given no evidence.

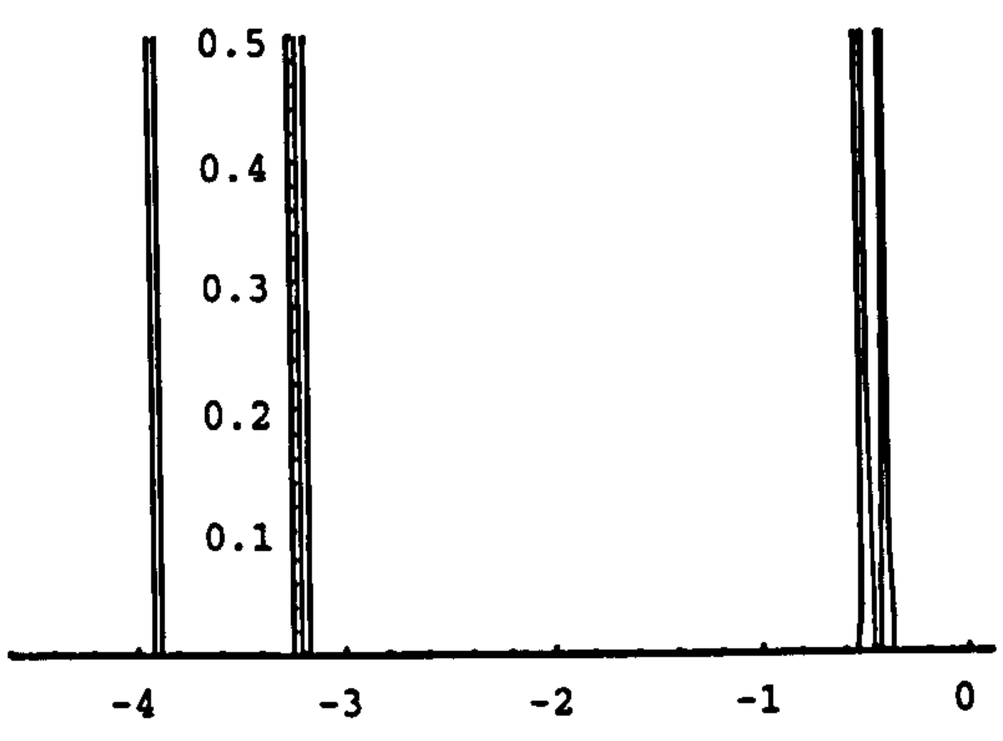
By comparing the graphs of the marginal distributions of the continuous random variables in Figure 4.10 with their means and variances in Table 4.9 we can see the obvious power of obtaining the probability density functions of the variables rather than just their moments. The first two moments of the distribution, while being useful, are not sufficient to characterise the distribution and can, if used blindly, give unexpected results. The mean of *Mi* (the amount of metal in the waste), for example, is at a value which is very unlikely to ever be observed. Similarly the moments of *E* (Filter Efficiency) give us no indication that *E* is formed from four Normal distributions with such small variances. The moments of CGM-distributions may thus be of limited value to the non-expert unless he has more insight into the problem.

Tables 4.10, 4.11, 4.12, and 4.13 give the marginal distributions of the discrete and continuous variables in the waste incinerator problem given that there is no evidence.

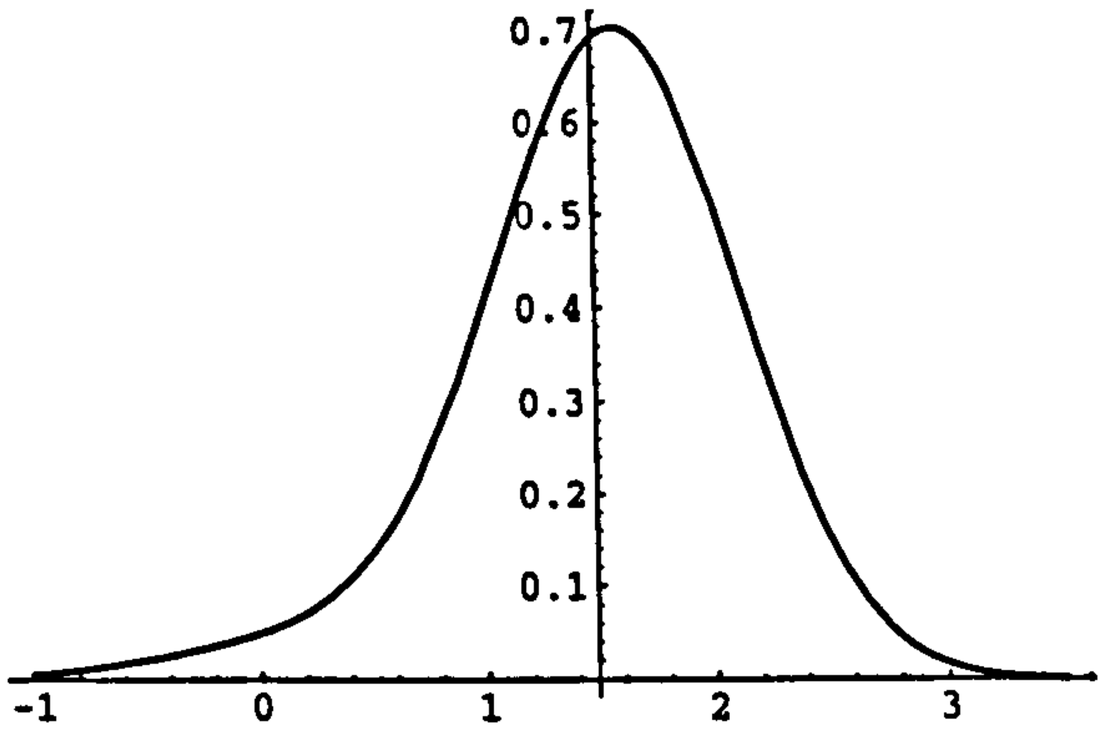



$C$ : $CO_2$ in emission

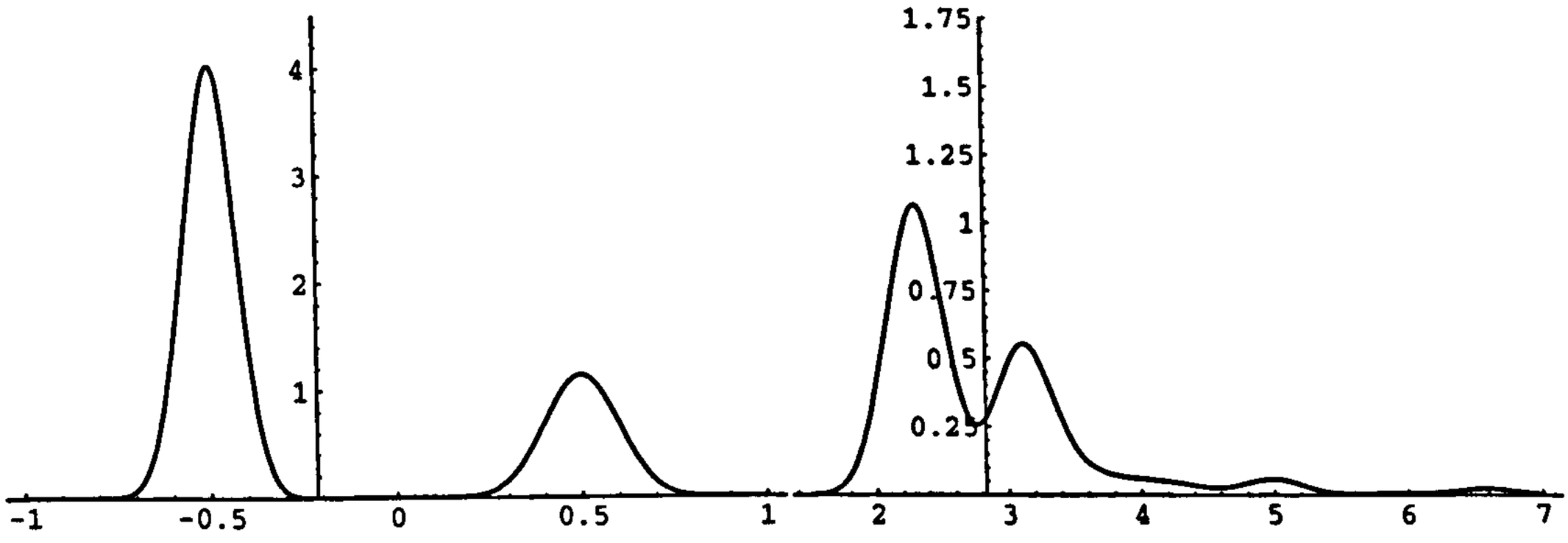
$D$ : Dust in emission




$E$ : Filter efficiency




$L$ : Light penetrability




$Mi$ : Metal in waste

$M0$ : Emission of metal

Figure 4.10: Graphs of the marginal distributions of the continuous variables in the waste incinerator problem given no evidence.

| B: Burning Regime | |
|---|---|
| B: Stable | $\frac{17}{20}$ |
| B: Unstable | $\frac{3}{20}$ |

| F: Filter State | |
|---|---|
| F: Intact | $\frac{19}{20}$ |
| F: Defective | $\frac{1}{20}$ |

| W: Type of Waste | |
|---|---|
| W: Industrial | $\frac{2}{7}$ |
| W: Household | $\frac{5}{7}$ |

Table 4.10: The marginal distributions of the discrete variables $B$, $F$ and $W$ in the waste incinerator problem given that there is no evidence.

| C: $CO_2$ in Emission |
|---|
| $\text{sqrt}\left[\frac{3}{80\Pi}\right] * \exp\left[\frac{-5}{3} - \frac{10c}{3} - \frac{5c^2}{3}\right]$ |
| $+ \quad \text{sqrt}\left[\frac{289}{80\Pi}\right] * \exp\left[-20 - 20c - 5c^2\right]$ |

| D: Emission of Dust |
|---|
| $\text{sqrt}\left[\frac{9}{392392\Pi}\right] * \exp\left[\frac{-252050}{1001} + \frac{71000d}{1001} - \frac{5000d^2}{1001}\right]$ |
| $+ \quad \text{sqrt}\left[\frac{225}{1569568\Pi}\right] * \exp\left[\frac{-16250}{77} + \frac{5000d}{77} - \frac{5000d^2}{1001}\right]$ |
| $+ \quad \text{sqrt}\left[\frac{289}{117992\Pi}\right] * \exp\left[\frac{-186050}{301} + \frac{61000d}{301} - \frac{5000d^2}{301}\right]$ |
| $+ \quad \text{sqrt}\left[\frac{5415}{653464\Pi}\right] * \exp\left[\frac{-108000}{1667} + \frac{60000d}{1667} - \frac{25000d^2}{5001}\right]$ |
| $+ \quad \text{sqrt}\left[\frac{7225}{628768\Pi}\right] * \exp\left[\frac{-151250}{401} + \frac{55000d}{401} - \frac{5000d^2}{401}\right]$ |
| $+ \quad \text{sqrt}\left[\frac{135375}{2613856\Pi}\right] * \exp\left[\frac{-361000}{5001} + \frac{190000d}{5001} - \frac{25000d^2}{5001}\right]$ |
| $+ \quad \text{sqrt}\left[\frac{27455}{30968\Pi}\right] * \exp\left[\frac{-169000}{1501} + \frac{130000d}{1501} - \frac{25000d^2}{1501}\right]$ |
| $+ \quad \text{sqrt}\left[\frac{13041125}{3137568\Pi}\right] * \exp\left[\frac{-196000}{2001} + \frac{140000d}{2001} - \frac{25000d^2}{2001}\right]$ |

Table 4.11: The marginal distributions of the continuous variables $C$ and $D$ in the waste incinerator problem given that there is no evidence.

**E: Filter Efficiency**

$$\text{sqrt}\left[\frac{50}{49\Pi}\right] * \exp\left[-800 - 4000e - 5000e^2\right]$$
$$+ \text{sqrt}\left[\frac{625}{98\Pi}\right] * \exp\left[-1250 - 5000e - 5000e^2\right]$$
$$+ \text{sqrt}\left[\frac{90250}{49\Pi}\right] * \exp\left[-380250 - 195000e - 25000e^2\right]$$
$$+ \text{sqrt}\left[\frac{1128125}{98\Pi}\right] * \exp\left[-256000 - 160000e - 25000e^2\right]$$

**L: Penetrability of Light**

$$\text{sqrt}\left[\frac{3}{359366\Pi}\right] * \exp\left[\frac{-6050}{11001} - \frac{22000l}{11001} - \frac{20000l^2}{11001}\right]$$
$$+ \text{sqrt}\left[\frac{75}{1437464\Pi}\right] * \exp\left[\frac{-1250}{11001} - \frac{10000l}{11001} - \frac{20000l^2}{11001}\right]$$
$$+ \text{sqrt}\left[\frac{289}{1009498\Pi}\right] * \exp\left[\frac{-50}{10301} - \frac{2000l}{10301} - \frac{20000l^2}{10301}\right]$$
$$+ \text{sqrt}\left[\frac{7225}{4077192\Pi}\right] * \exp\left[\frac{-1250}{10401} + \frac{10000l}{10401} - \frac{20000l^2}{10401}\right]$$
$$+ \text{sqrt}\left[\frac{16245}{5390098\Pi}\right] * \exp\left[\frac{-144000}{55001} + \frac{240000l}{55001} - \frac{100000l^2}{55001}\right]$$
$$+ \text{sqrt}\left[\frac{406125}{21560392\Pi}\right] * \exp\left[\frac{-121000}{55001} + \frac{220000l}{55001} - \frac{100000l^2}{55001}\right]$$
$$+ \text{sqrt}\left[\frac{521645}{5047098\Pi}\right] * \exp\left[\frac{-289000}{51501} - \frac{340000l}{51501} - \frac{100000l^2}{51501}\right]$$
$$+ \text{sqrt}\left[\frac{13041125}{20384392\Pi}\right] * \exp\left[\frac{-256000}{52001} - \frac{320000l}{52001} - \frac{100000l^2}{52001}\right]$$

Table 4.12: The marginal distributions of the continuous variables $E$ and $L$ in the waste incinerator problem given that there is no evidence.

**M0: Emission of Metal**

$$\text{sqrt}\left[\frac{9}{439432\Pi}\right] * \exp\left[\frac{-15200}{59} + \frac{4000m0}{59} - \frac{5000m0^2}{1121}\right]$$

$$+ \text{sqrt}\left[\frac{25}{186592\Pi}\right] * \exp\left[\frac{-20000}{119} + \frac{20000m0}{357} - \frac{5000m0^2}{1071}\right]$$

$$+ \text{sqrt}\left[\frac{289}{165032\Pi}\right] * \exp\left[\frac{-217800}{421} + \frac{66000m0}{421} - \frac{5000m0^2}{421}\right]$$

$$+ \text{sqrt}\left[\frac{5415}{731864\Pi}\right] * \exp\left[\frac{-420250}{5601} + \frac{205000m0}{5601} - \frac{25000m0^2}{5601}\right]$$

$$+ \text{sqrt}\left[\frac{7225}{738528\Pi}\right] * \exp\left[\frac{-125000}{471} + \frac{50000m0}{471} - \frac{5000m0^2}{471}\right]$$

$$+ \text{sqrt}\left[\frac{406125}{8390368\Pi}\right] * \exp\left[\frac{-272250}{5351} + \frac{165000m0}{5351} - \frac{25000m0^2}{5351}\right]$$

$$+ \text{sqrt}\left[\frac{521645}{823592\Pi}\right] * \exp\left[\frac{-240250}{2101} + \frac{155000m0}{2101} - \frac{25000m0^2}{2351}\right]$$

$$+ \text{sqrt}\left[\frac{13041125}{3686368\Pi}\right] * \exp\left[\frac{-132250}{2351} + \frac{115000m0}{2351} - \frac{25000m0^2}{2351}\right]$$

**Mi: Metal in Waste**

$$\text{sqrt}\left[\frac{200}{49\Pi}\right] * \exp\left[\frac{-25}{2} + 50mi - 50mi^2\right]$$

$$+ \text{sqrt}\left[\frac{2500}{49\Pi}\right] * \exp\left[-25 - 100mi - 100mi^2\right]$$

Table 4.13: The marginal distributions of the continuous variables $M0$ and $Mi$ in the waste incinerator problem given that there is no evidence.

## 4.11 Adding Evidence

We shall now consider how we may add a set of evidence $\mathcal{E}$ to our system. We shall assume that evidence on a discrete variable $I_U$ informs us that $I_U$ is at some given state $i_U^*$, say, and we wish to update our uncertainty appropriately. We do this by removing all cells that have been deemed impossible by the evidence - i.e. the cells for which $i_U \neq i_U^*$. This reduces the dimension of the potential tables by one for each discrete variable thus simplifying the computational complexity of our problem. Such evidence must be entered into every universe containing the discrete variable $I_U$.

Evidence on a continuous variable $Y_U$ is assumed to indicate that $Y_U$ is at a particular level $y_U^*$, say. In order to enter this evidence into the system we must replace every occurrence of the symbolic variable $y_U$ in every potential function by the numeric evidence $y_U^*$. This will be required in every universe containing the continuous variable $Y_U$.

Having entered a collection of evidence $\mathcal{E}$ into the system in our *Mathematica* implementation we update `fullctsvars`, `fulldisvars` and `numdisvars` to inform the system that the dimensions of the potential tables have changed and the known variables are there in name only. We then pass a propagation schedule. This results in the joint system belief being proportional to the joint probability density function of the variables given the evidence $\mathcal{E}$. If we marginalise any clique or separator over the variables that it contains we obtain the normalisation constant required to obtain the updated joint system belief and hence the marginals.

Lauritzen (1992) considers the addition of the evidence $\mathcal{E}$ : $\{W = \text{Industrial}, L = 1.1, \text{and } C = -0.9\}$ into the system while Olesen (1991) considers the addition of the evidence $\mathcal{E}$ : $\{W = \text{Industrial}, L = 0.5, \text{and } C = -1.6\}$ into the system. For completeness the results of both of these collections of evidence are presented here. The marginal distributions of the unknown variables given Lauritzen's evidence are given in Tables 4.14 and 4.15, they are plotted in Figure 4.11, and their moments are given in Table 4.17. The normalising constant given Lauritzen's evidence is in Table 4.16. The marginal distributions of the unknown variables given Olesen's evidence are given in Tables 4.18 and 4.19, they are plotted in Figure 4.12, and their moments are given in Table 4.21. The normalising constant given Olesen's evidence is in Table 4.20.

| B: Burning Regime | |
|---|---|
| B: Stable | $\lambda\Big($ $\exp[\frac{-1775421}{206020}] * \mathrm{sqrt}[\frac{1445}{1009498*P_i^2}]$ $+ \exp[\frac{-2317207}{343340}] * \mathrm{sqrt}[\frac{2608225}{5047098*P_i^2}]\Big) = 0.012253$ |
| B: Unstable | $\lambda\Big($ $\exp[\frac{-1092667}{220020}] * \mathrm{sqrt}[\frac{5}{359366*P_i^2}]$ $+ \exp[\frac{-115001}{3300060}] * \mathrm{sqrt}[\frac{27075}{5390098*P_i^2}]\Big) = 0.987747$ |

| F: Filter State | |
|---|---|
| F: Intact | $\lambda\Big($ $\exp[\frac{-2317207}{343340}] * \mathrm{sqrt}[\frac{2608225}{5047098*P_i^2}]$ $+ \exp[\frac{-115001}{3300060}] * \mathrm{sqrt}[\frac{27075}{5390098*P_i^2}]\Big) = 0.999526$ |
| F: Defective | $\lambda\Big($ $\exp[\frac{-1775421}{206020}] * \mathrm{sqrt}[\frac{1445}{1009498*P_i^2}]$ $+ \exp[\frac{-1092667}{220020}] * \mathrm{sqrt}[\frac{5}{359366*P_i^2}]\Big) = 0.000474$ |

| D: Emission of Dust |
|---|
| $\lambda\Big($ $\exp[\frac{-77788171}{300300} + \frac{374019*d}{5005} - \frac{11001*d^2}{2002}] * \mathrm{sqrt}[\frac{15}{196196*P_i^3}]$ $+ \exp[\frac{-19004427}{30100} + \frac{310719*d}{1505} - \frac{10301*d^2}{602}] * \mathrm{sqrt}[\frac{1445}{58996*P_i^3}]$ $+ \exp[\frac{-36019057}{500100} + \frac{331673*d}{8335} - \frac{55001*d^2}{10002}] * \mathrm{sqrt}[\frac{9025}{326732*P_i^3}]$ $+ \exp[\frac{-18891827}{150100} + \frac{678519*d}{7505} - \frac{51501*d^2}{3002}] * \mathrm{sqrt}[\frac{137275}{15484*P_i^3}]\Big)$ |

| E: Filter Efficiency |
|---|
| $\lambda\Big($ $\exp[\frac{-266059}{330} - \frac{44037*e}{11} - \frac{55005*e^2}{11}] * \mathrm{sqrt}[\frac{75}{1078*P_i^3}]$ $+ \exp[\frac{-1667753}{2060} - \frac{412270*e}{103} - \frac{515050*e^2}{103}] * \mathrm{sqrt}[\frac{36125}{5047*P_i^3}]$ $+ \exp[\frac{-125484559}{330} - \frac{2145037*e}{11} - \frac{275005*e^2}{11}] * \mathrm{sqrt}[\frac{135375}{1078*P_i^3}]$ $+ \exp[\frac{-783334753}{2060} - \frac{20085270*e}{103} - \frac{2575050*e^2}{103}] * \mathrm{sqrt}[\frac{65205625}{5047*P_i^3}]\Big)$ |

Table 4.14: The marginal distributions of the variables in the waste incinerator problem given Lauritzen's evidence $W$ = Industrial, $C = -0.9$ , $L = 1.1$

**Mi: Metal in Waste**

$$\lambda\Big( \exp[\tfrac{-3842917}{220020} + 50 * mi - 50 * mi^2] * \text{sqrt}[\tfrac{125}{179683 * Pi^3}]$$
$$+ \exp[\tfrac{-4350671}{206020} + 50 * mi - 50 * mi^2] * \text{sqrt}[\tfrac{36125}{504749 * Pi^3}]$$
$$+ \exp[\tfrac{-41365751}{3300060} + 50 * mi - 50 * mi^2] * \text{sqrt}[\tfrac{676875}{2695049 * Pi^3}]$$
$$+ \exp[\tfrac{-6608957}{343340} + 50 * mi - 50 * mi^2] * \text{sqrt}[\tfrac{65205625}{2523549 * Pi^3}]\Big)$$

**M0: Emission of Metal**

$$\lambda\Big( \exp[\tfrac{-1120224482}{4248795} + \tfrac{20076075 * m0}{283253} - \tfrac{1375125 * m0^2}{283253}] * \text{sqrt}[\tfrac{1875}{27758794 * Pi^3}]$$
$$+ \exp[\tfrac{-558672619}{1061530} + \tfrac{16823575 * m0}{106153} - \tfrac{1287625 * m0^2}{106153}] * \text{sqrt}[\tfrac{180625}{10402994 * Pi^3}]$$
$$+ \exp[\tfrac{-194412748}{2358755} + \tfrac{56626075 * m0}{1415253} - \tfrac{6875125 * m0^2}{1415253}] * \text{sqrt}[\tfrac{59375}{2433242 * Pi^3}]$$
$$+ \exp[\tfrac{-95492417}{756790} + \tfrac{5766225 * m0}{75679} - \tfrac{6437625 * m0^2}{529753}] * \text{sqrt}[\tfrac{326028125}{51915794 * Pi^3}]\Big)$$

Table 4.15: The marginal distributions of the variables in the waste incinerator problem given Lauritzen's evidence $W =$ Industrial, $C = -0.9$ , $L = 1.1$

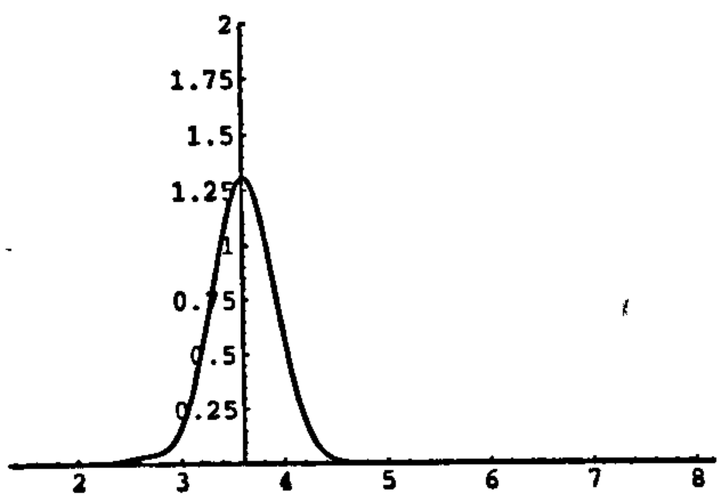
**$\lambda$: Normalising Constant**

$$\begin{aligned}
\tfrac{1}{\lambda} &= \exp[\tfrac{-1775421}{206020}] * \text{sqrt}[\tfrac{1445}{1009498 * Pi^2}] + \exp[\tfrac{-2317207}{343340}] * \text{sqrt}[\tfrac{2608225}{5047098 * Pi^2}] \\
&+ \exp[\tfrac{-1092667}{220020}] * \text{sqrt}[\tfrac{5}{359366 * Pi^2}] + \exp[\tfrac{-115001}{3300060}] * \text{sqrt}[\tfrac{27075}{5390098 * Pi^2}] \\
&= P(\text{Evidence}) \\
&= 0.022066
\end{aligned}$$

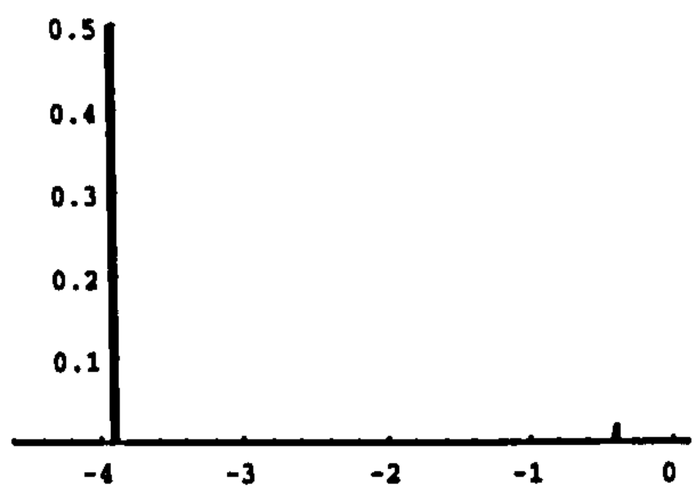
Table 4.16: The normalising constant given Lauritzen's evidence $W =$ Industrial, $C = -0.9$ , $L = 1.1$

| Variable | Mean | Variance |
|---|---|---|
| D: Emission of Dust | 3.607667 | 0.106179 |
| E: Filter Efficiency | -3.898338 | 0.005819 |
| Mi: Metal in Waste | 0.500000 | 0.010000 |
| M0: Emission of Metal | 4.107667 | 0.118179 |

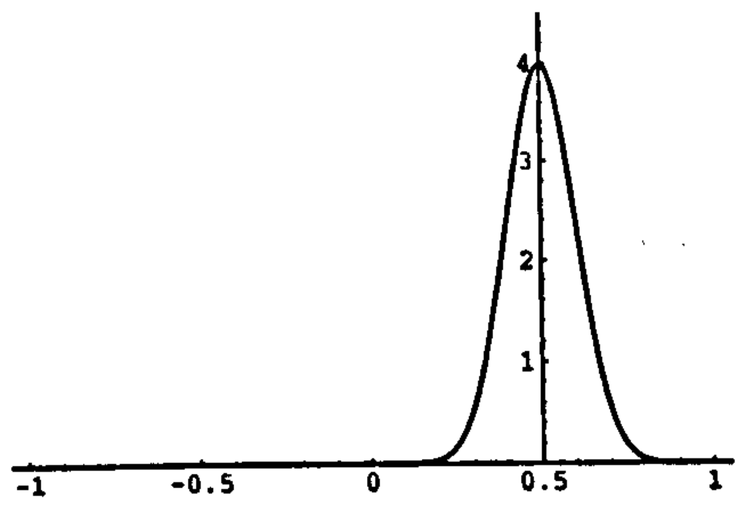
Table 4.17: Means and variances of continuous variables given Lauritzen's evidence $W =$ Industrial, $C = -0.9$ , $L = 1.1$



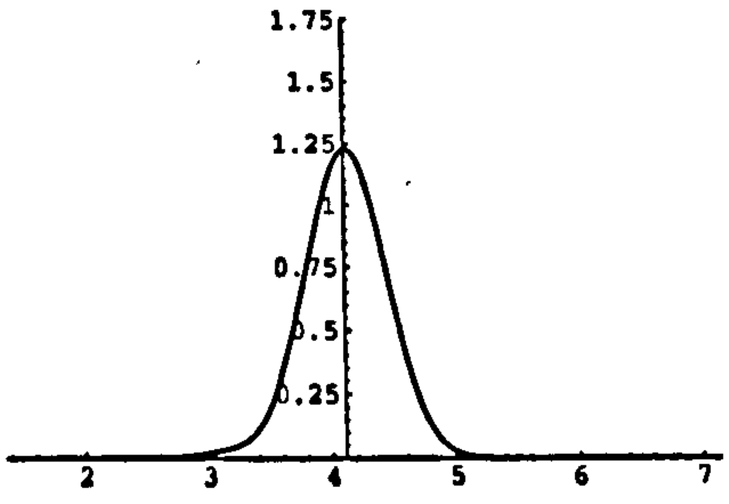

$D$ : Dust in emission




$E$ : Filter efficiency

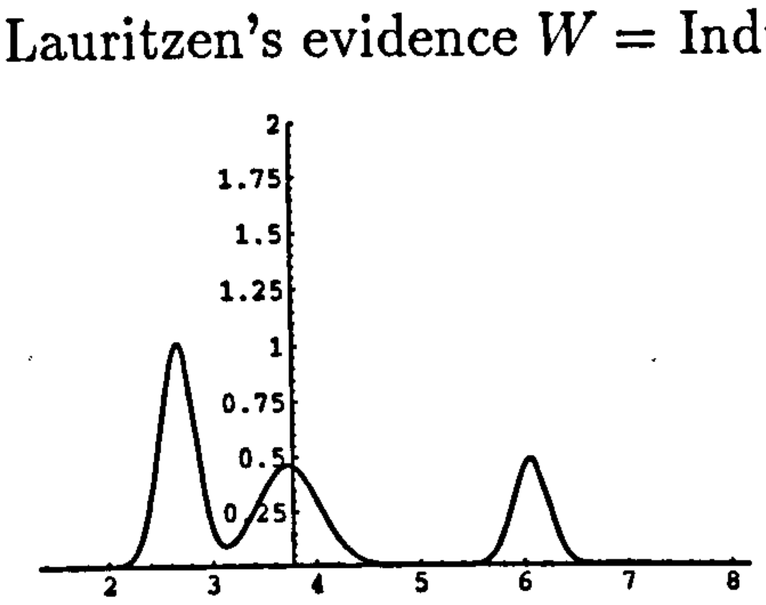



$Mi$ : Metal in waste

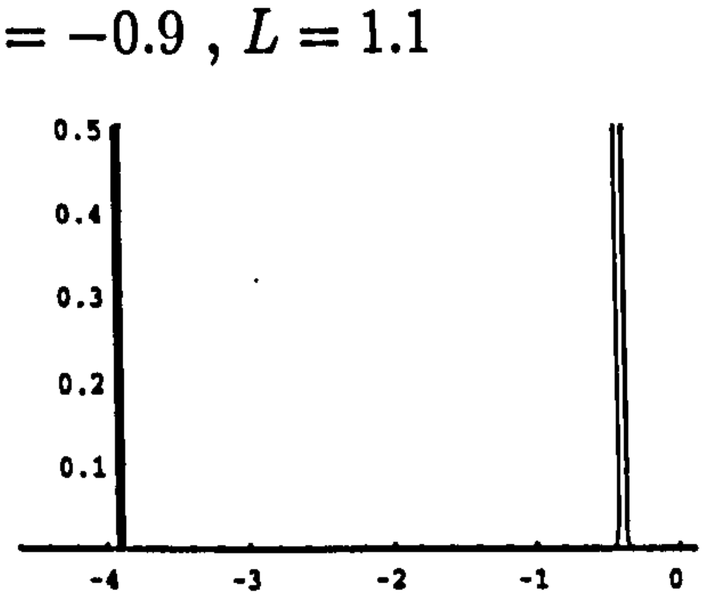



$M0$ : Emission of metal

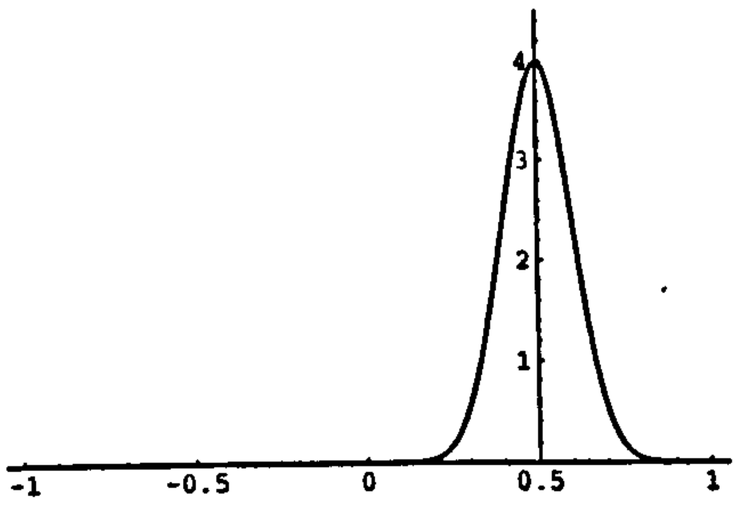
Figure 4.11: Graphs of the marginal distributions in the waste incinerator problem given Lauritzen's evidence $W =$ Industrial, $C = -0.9$ , $L = 1.1$



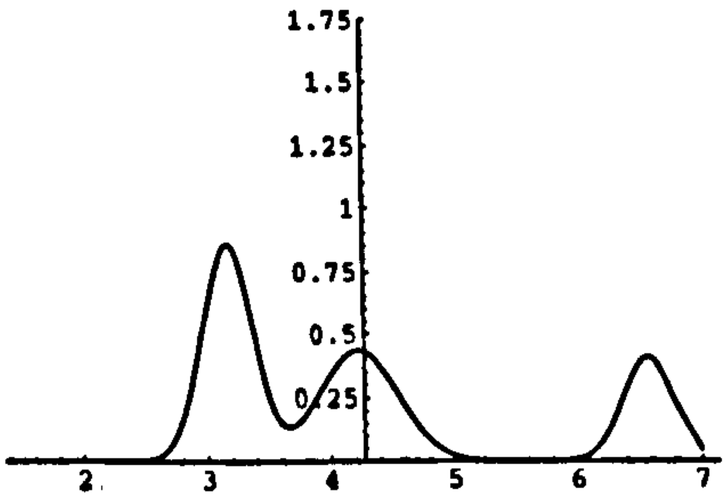

$D$ : Dust in emission




$E$ : Filter efficiency




$Mi$ : Metal in waste




$M0$ : Emission of metal

Figure 4.12: Graphs of the marginal distributions in the waste incinerator problem given Olesen's evidence $W =$ Industrial, $C = -1.6$ , $L = 0.5$

| B: Burning Regime | |
|---|---|
| B: Stable | $\lambda\Big($ $\exp[\frac{-71454}{51505}] * \mathrm{sqrt}[\frac{1445}{1009498*P_i^2}]$ $+ \exp[\frac{-308668}{85835}] * \mathrm{sqrt}[\frac{2608225}{5047098*P_i^2}]\Big) = 0.642434$ |
| B: Unstable | $\lambda\Big($ $\exp[\frac{-47751}{18335}] * \mathrm{sqrt}[\frac{5}{359366*P_i^2}]$ $+ \exp[\frac{-410003}{275005}] * \mathrm{sqrt}[\frac{27075}{5390098*P_i^2}]\Big) = 0.357566$ |

| F: Filter State | |
|---|---|
| F: Intact | $\lambda\Big($ $\exp[\frac{-308668}{85835}] * \mathrm{sqrt}[\frac{2608225}{5047098*P_i^2}]$ $+ \exp[\frac{-410003}{275005}] * \mathrm{sqrt}[\frac{27075}{5390098*P_i^2}]\Big) = 0.785816$ |
| F: Defective | $\lambda\Big($ $\exp[\frac{-47751}{18335}] * \mathrm{sqrt}[\frac{5}{359366*P_i^2}]$ $+ \exp[\frac{-71454}{51505}] * \mathrm{sqrt}[\frac{1445}{1009498*P_i^2}]\Big) = 0.214184$ |

| D: Emission of Dust |
|---|
| $\lambda\Big($ $\exp[\frac{-2651631}{10010} + \frac{76005*d}{1001} - \frac{11001*d^2}{2002}] * \mathrm{sqrt}[\frac{15}{196196*P_i^3}]$ $+ \exp[\frac{-1900533}{3010} + \frac{62505*d}{301} - \frac{10301*d^2}{602}] * \mathrm{sqrt}[\frac{1445}{58996*P_i^3}]$ $+ \exp[\frac{-1298377}{16670} + \frac{68335*d}{1667} - \frac{55001*d^2}{10002}] * \mathrm{sqrt}[\frac{9025}{326732*P_i^3}]$ $+ \exp[\frac{-1889633}{15010} + \frac{137505*d}{1501} - \frac{51501*d^2}{3002}] * \mathrm{sqrt}[\frac{137275}{15484*P_i^3}]\Big)$ |

| E: Filter Efficiency |
|---|
| $\lambda\Big($ $\exp[\frac{-176757}{220} - \frac{44025*e}{11} - \frac{55005*e^2}{11}] * \mathrm{sqrt}[\frac{75}{1078*P_i^3}]$ $+ \exp[\frac{-825949}{1030} - \frac{412150*e}{103} - \frac{515050*e^2}{103}] * \mathrm{sqrt}[\frac{36125}{5047*P_i^3}]$ $+ \exp[\frac{-83655757}{220} - \frac{2145025*e}{11} - \frac{275005*e^2}{11}] * \mathrm{sqrt}[\frac{135375}{1078*P_i^3}]$ $+ \exp[\frac{-391659449}{1030} - \frac{20085150*e}{103} - \frac{2575050*e^2}{103}] * \mathrm{sqrt}[\frac{65205625}{5047*P_i^3}]\Big)$ |

Table 4.18: The marginal distributions of the variables in the waste incinerator problem given Olesen's evidence $W$ = Industrial, $C = -1.6$ , $L = 0.5$

**Mi: Metal in Waste**

$$\lambda\Big( \exp[\tfrac{-553877}{36670} + 50 * mi - 50 * mi^2] * \text{sqrt}[\tfrac{125}{179683 * Pi^3}]$$
$$+ \exp[\tfrac{-1430533}{103010} + 50 * mi - 50 * mi^2] * \text{sqrt}[\tfrac{36125}{504749 * Pi^3}]$$
$$+ \exp[\tfrac{-7695131}{550010} + 50 * mi - 50 * mi^2] * \text{sqrt}[\tfrac{676875}{2695049 * Pi^3}]$$
$$+ \exp[\tfrac{-2763211}{171670} + 50 * mi - 50 * mi^2] * \text{sqrt}[\tfrac{65205625}{2523549 * Pi^3}]\Big)$$

**M0: Emission of Metal**

$$\lambda\Big( \exp[\tfrac{-1524422661}{5665060} + \tfrac{20376375 * m0}{283253} - \tfrac{1375125 * m0^2}{283253}] * \text{sqrt}[\tfrac{1875}{27758794 * Pi^3}]$$
$$+ \exp[\tfrac{-1113824573}{2123060} + \tfrac{16913875 * m0}{106153} - \tfrac{1287625 * m0^2}{106153}] * \text{sqrt}[\tfrac{180625}{10402994 * Pi^3}]$$
$$+ \exp[\tfrac{-2499373661}{28305060} + \tfrac{58126375 * m0}{1415253} - \tfrac{6875125 * m0^2}{1415253}] * \text{sqrt}[\tfrac{59375}{2433242 * Pi^3}]$$
$$+ \exp[\tfrac{-1331879173}{10595060} + \tfrac{40813875 * m0}{529753} - \tfrac{6437625 * m0^2}{529753}] * \text{sqrt}[\tfrac{326028125}{51915794 * Pi^3}]\Big)$$

Table 4.19: The marginal distributions of the variables in the waste incinerator problem given Olesen's evidence $W$ = Industrial, $C = -1.6$ , $L = 0.5$

**$\lambda$: Normalising Constant**

$$\begin{aligned}
\tfrac{1}{\lambda} &= \exp[\tfrac{-71454}{51505}] * \text{sqrt}[\tfrac{1445}{1009498 * Pi^2}] + \exp[\tfrac{-308668}{85835}] * \text{sqrt}[\tfrac{2608225}{5047098 * Pi^2}] \\
&+ \exp[\tfrac{-47751}{18335}] * \text{sqrt}[\tfrac{5}{359366 * Pi^2}] + \exp[\tfrac{-410003}{275005}] * \text{sqrt}[\tfrac{27075}{5390098 * Pi^2}] \\
&= P(\text{Evidence}) \\
&= 0.014452
\end{aligned}$$

Table 4.20: The normalising constant given Olesen's evidence $W$ = Industrial, $C = -1.6$ , $L = 0.5$

| Variable | Mean | Variance |
|---|---|---|
| D: Emission of Dust | 3.774482 | 1.736158 |
| E: Filter Efficiency | -3.150352 | 2.061649 |
| Mi: Metal in Waste | 0.500000 | 0.010000 |
| M0: Emission of Metal | 4.274482 | 1.748158 |

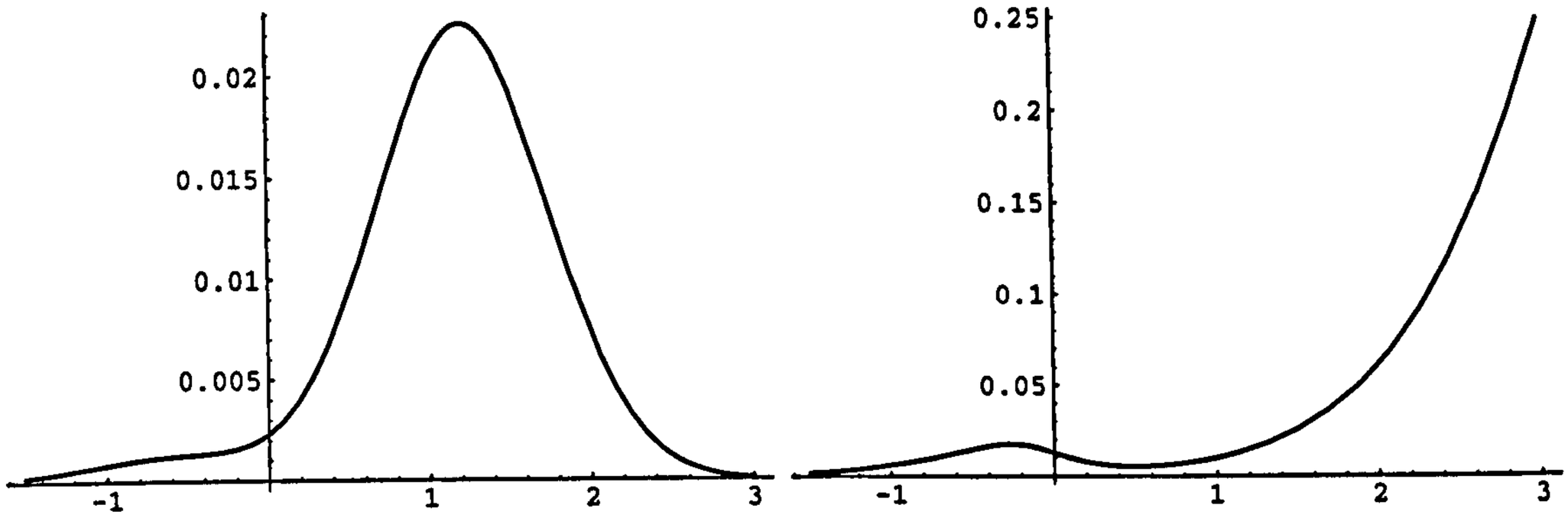
Table 4.21: Means and variances of continuous variables given Olesen's evidence $W$ = Industrial, $C = -1.6$ , $L = 0.5$

## 4.12 Adding Symbolic Evidence

Not only does our symbolic implementation allow us to enter numeric evidence on a continuous variable but we may, with no extra programming, enter symbolic evidence on any variable. For example, suppose we have the evidence that $W =$ Industrial and $C = -0.9$ then we might wish to investigate how the other variables will respond to the penetrability of light, $L$, being changed. We thus enter the additional evidence that $L = \mathcal{L}$, where $\mathcal{L}$ is a symbolic variable, and propagate this evidence through the system. The resulting normalising constant is thus dependent on $L$ and is plotted in Figure 4.13.

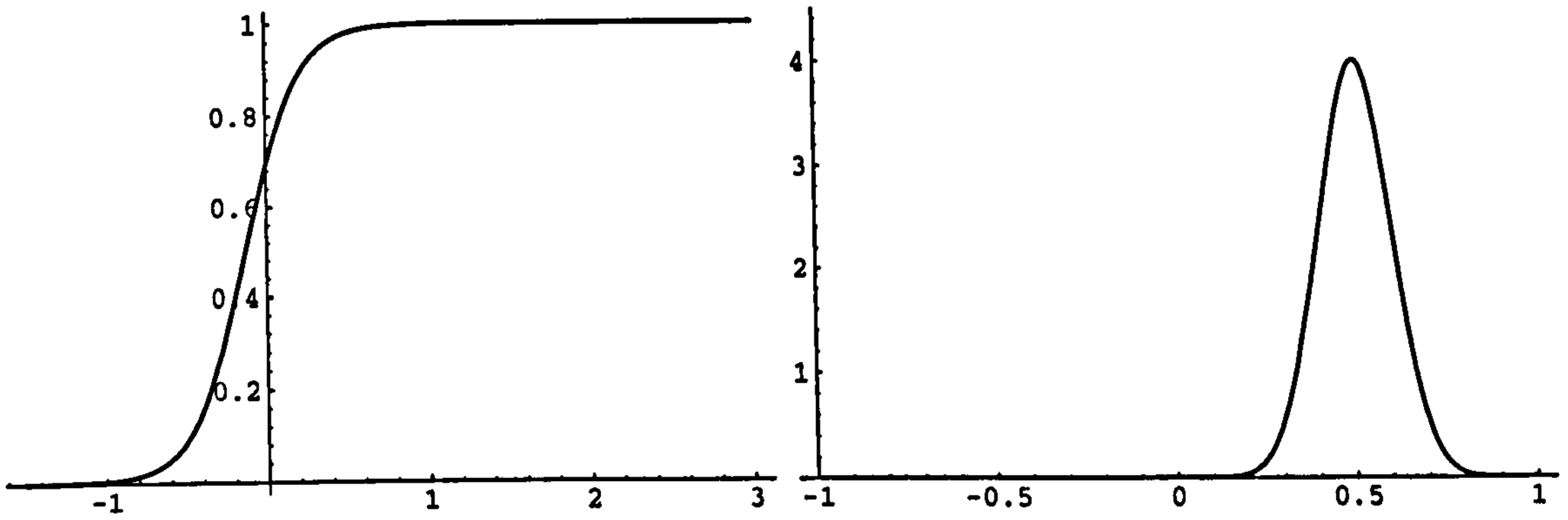
Normalising the potentials we find that all the variables vary with respect to $L$ except for $Mi$, the amount of metal in the waste (see Figure 4.13). Considering the CPN in Figure 4.8, the reason for this is obvious since $Mi$'s only ancestor is $W$, the type of waste, which is defined to be industrial. The probabilities that the burning regime, $B$, is stable and the filter state, $F$, is intact are given for varying levels of $L$ in Figure 4.13. The means and variances of the Filter Efficiency, $E$, the emission of dust, $D$, and the emission of metal, $M0$, are given in Figures 4.13, 4.14, and 4.15 respectively. Plots of the marginal distributions of $D$ and $M0$, for given values of $L$, are also presented in Figures 4.14 and 4.15. Arranging such plots in a group in *Mathematica* enables us to animate the plots providing a dynamic way to investigate how dust and metal emissions are affected by changing light penetrability.

The advantage of being able to add symbolic evidence is that only one propagation schedule need be passed to investigate the effects of an infinite number of different observations. This may be very useful especially in terms of gaining a graphical representation of a problem. The trade-off is that the addition of symbolic evidence on "too many" continuous variables is likely to be computationally unwieldy and the structural niceties of the PES and propagation algorithm will be lost.



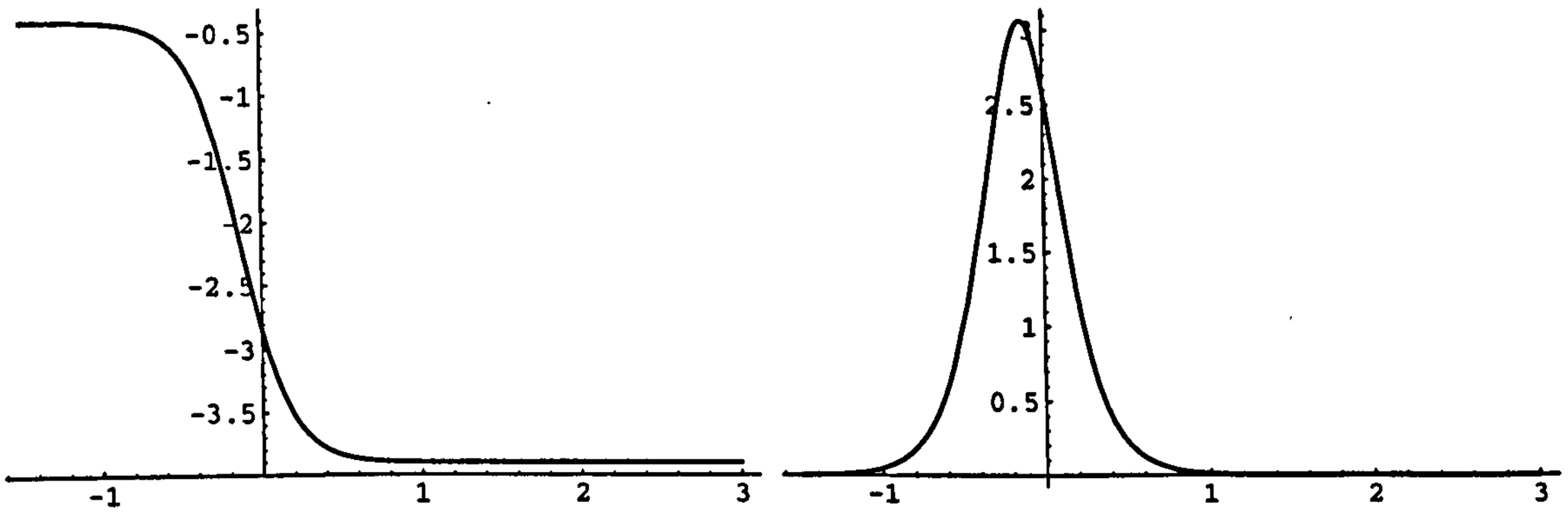

Normalising constant given
$L = -1.5$ to 3.0

P(Burning Regime = Stable) given
$L = -1.5$ to 3.0




P(Filter = Intact) given
$L = -1.5$ to 3.0

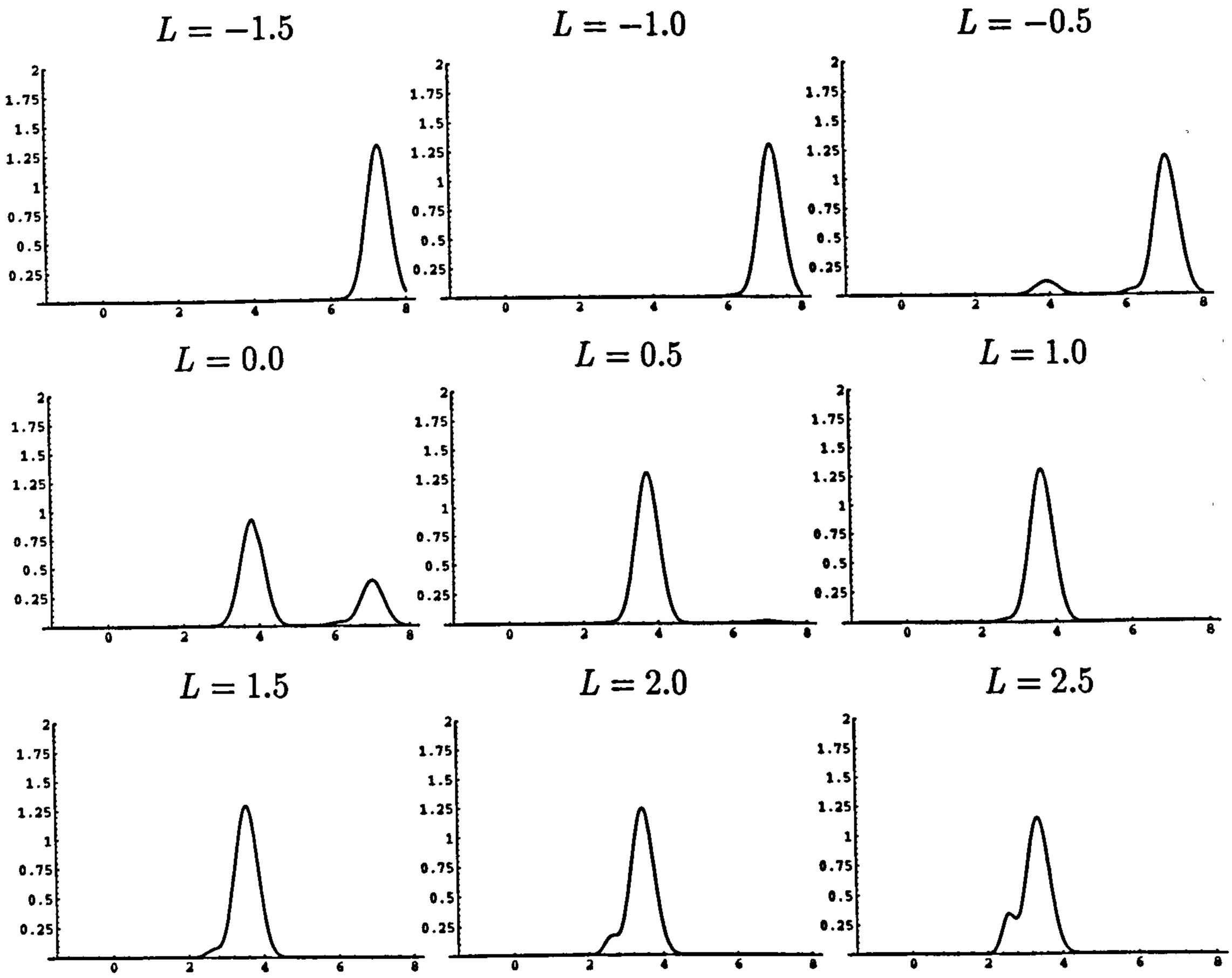
$Mi$ : Metal in waste



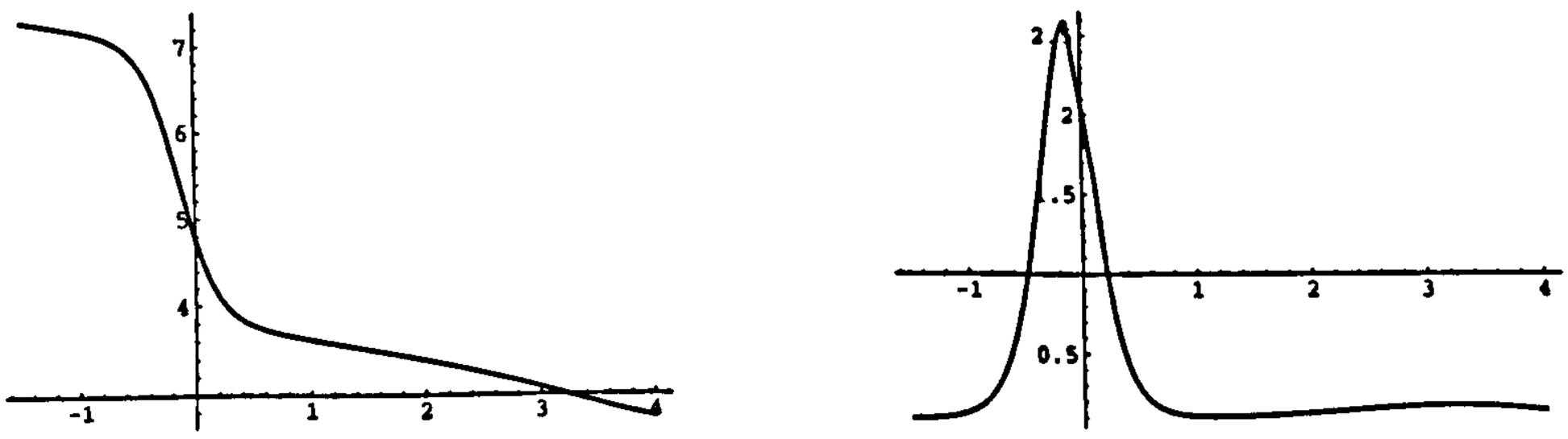

Mean of $E$ : Filter efficiency given
$L = -1.5$ to 3.0

Variance of $E$ : Filter efficiency given
$L = -1.5$ to 3.0

Figure 4.13: Probabilities, means, variances, and probability density functions of the marginal distributions of variables in the waste incinerator problem given evidence $W =$ Industrial, $C = -0.9$ and symbolic evidence on $L$.

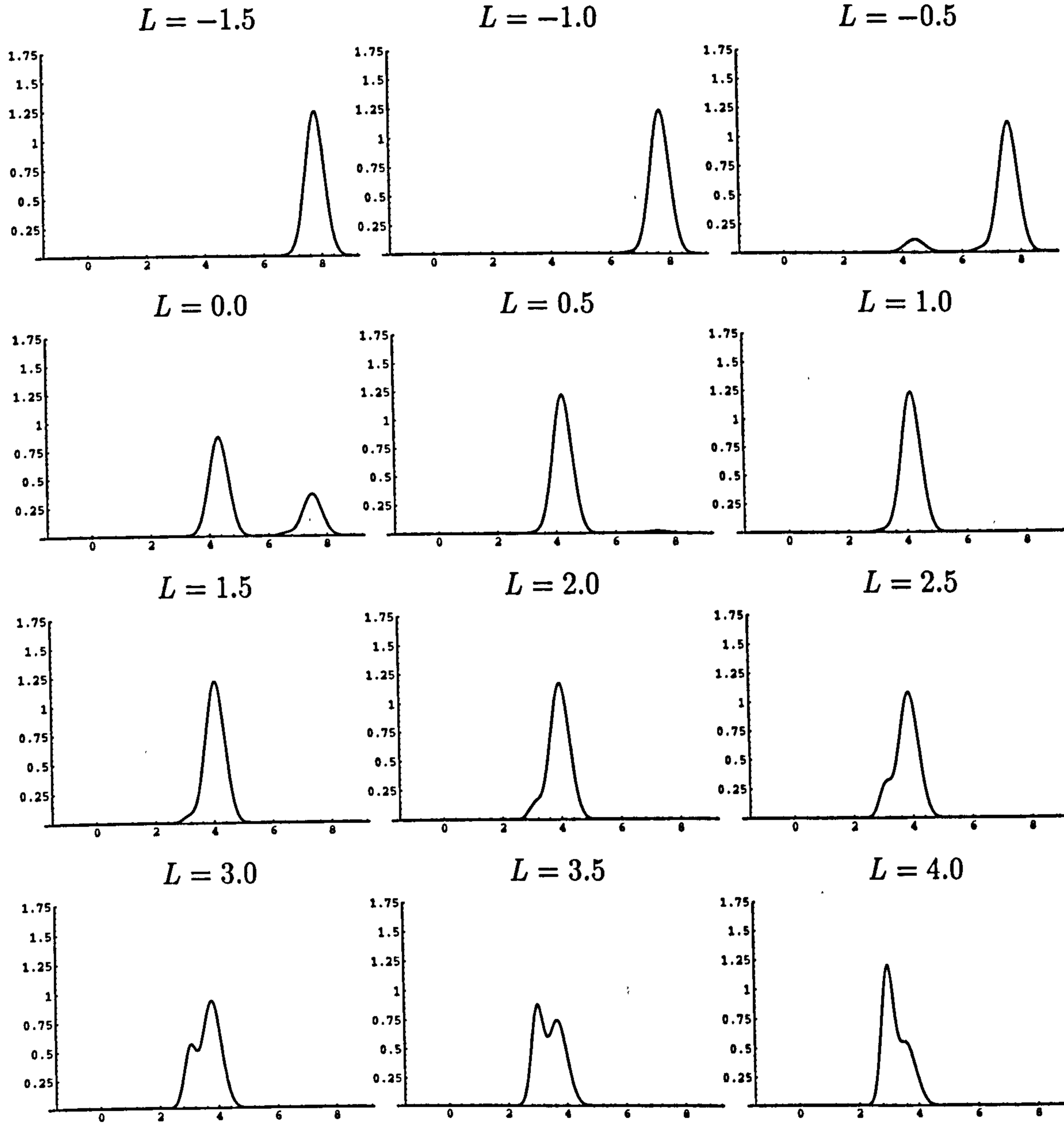



Plot of distribution of $D$ for evidence $L = -1.5$ to 4.0



Mean of $D$ given $L = -1.5$ to 4.0 Variance of $D$ given $L = -1.5$ to 4.0

Figure 4.14: Mean, variance and probability density functions of the marginal distribution of $D$ (emission of dust) in the waste incinerator problem given evidence $W =$ Industrial, $C = -0.9$ and symbolic evidence on $L$.




Plot of distribution of $M0$ for evidence $L = -1.5$ to 4.0



Mean of $M0$ given $L = -1.5$ to 4.0



Variance of $M0$ given $L = -1.5$ to 4.0

Figure 4.15: Mean, variance and probability density functions of the marginal distribution of $M0$ (emission of metal) in the waste incinerator problem given evidence $W =$ Industrial, $C = -0.9$ and symbolic evidence on $L$.

## 4.13 Discussion

The marginal distributions of the continuous variables formed in our system are CGM-distributions. Such distributions may be expressed as the weighted sum of a series of Normal distributions. Suppose a CGM-distribution may be expressed as a weighted sum of $r$ Normal distributions then a total of $3r - 1$ parameters are required to describe it fully. These parameters correspond to the mean, variance, and weight attributed to each of the $r$ Normal distributions less one since the weights sum to one. In the waste incinerator problem, for example, the marginal distribution of $L$ (light penetrability) given no evidence, is formed as a weighted sum of eight Normal distributions. We will thus need a total of 23 parameters to describe it. In contrast Lauritzen's approach to the problem, as described in Chapter 3, gives us only two statistics - the mean and variance of the marginal distribution. Such an approach will thus only be sufficient for continuous variables with either no discrete ancestors, or no unknown discrete ancestors.

Apart from providing the full parameterisation of the continuous variables our method has several other advantages over Lauritzen's. The junction tree required is formed through a process of weak triangulation in contrast to the strong triangulation required by Lauritzen's scheme. In general, such junction trees have smaller cliques than their strongly triangulated counterparts. We are also able to gain a graphical output from our system without the need for simulation studies. This is very useful for the non-expert not only because he may not be particularly adept at interpreting formulae, but also because any peculiarities and pitfalls which might exist in the distribution and may otherwise be overlooked are made clear. After all, a picture paints a thousand words.

The disadvantage with our approach is that it may become both space hungry and slow when dealing with large networks. Consider, for example, a continuous variable with a large number of discrete ancestors or one whose discrete ancestors each take a large number of different states. The potentials pertaining to such a continuous variable will thus be very complex, take a long time to calculate and occupy a lot of space. They will, however, be accurate. A choice must thus be made by the expert system builder as to whether he requires speed or accuracy. Only then may he decide which of the two methods he wishes to use. It should also be noted that if he is only interested in the states of the discrete variables then Lauritzen's approach is clearly the superior method to use.

Recall that in order to be faithful to Lauritzen's approach we defined the conditional distribution of any continuous variable $Y_a$ in the system given its parents $X_{pa(a)} = (I_{pa(k)}, Y_{pa(k)})$ to be of the following type:

$$f_{Y_a|(I_{pa(a)},Y_{pa(a)})} = \frac{1}{\sqrt{2\Pi\gamma(i_{pa(a)})}} \exp\left[\frac{-1}{2\gamma(i_{pa(a)})}\left(y_a - \alpha(i_{pa(a)}) - \beta(i_{pa(a)})^T y_{pa(a)}\right)^2\right]$$

In other words $Y_a \mid X_{pa(a)}$ takes a conditional Normal distribution with mean $\alpha(i_{pa(a)}) + \beta(i_{pa(a)})^T y_{pa(a)}$ and variance $\gamma(i_{pa(a)})$. It should be noted that with our symbolic approach we are equally justified in defining the conditional distribution of any continuous variable $Y_a$ as a weighted mixture of conditional Normal distributions thus:

$$\begin{aligned} f_{Y_a|(I_{pa(a)},Y_{pa(a)})} &= \sum_{j=1}^{n} \frac{w(i_{pa(a)}, j)}{\sqrt{2\Pi\gamma(i_{pa(a)}, j)}} \\ &\times \exp\left[\frac{-1}{2\gamma(i_{pa(a)}, j)}\left(y_k - \alpha(i_{pa(a)}, j) - \beta(i_{pa(a)}, j)^T y_{pa(a)}\right)^2\right] \end{aligned}$$

Then, since the potentials in our system will remain to be CGM-potentials the techniques of our symbolic approach will continue to work.

## 4.14 The Sick Fish Problem

We have currently only investigated those discrete distributions which may be constrained to a *finite* table of numeric or symbolic probabilities. Many discrete distributions which take *infinite* ranges may be approximated using our current methodologies by truncating their distributions to fit finite ranges. In this section, however, we introduce an example of how a discrete variable with *nonfinite* range may be handled using exact symbolic methods. To facilitate this we shall consider a fictitious example composed of a set of discrete variables some of which are of finite range, others of which are not.

Let us introduce the "Sick Fish Problem". A fairground worker who runs a hoopla stall gives goldfish away as prizes. Unfortunately his fish are prone to be infected with an exoparasite which is visible as a white spot on the scales of the fish. If there are too many white spots, and hence parasites, on a fish, he will be unable to give it as a prize. The parasite is also harmful and may result in the death of the fish where, again, the fairground worker will be unable to give it as a prize. He routinely treats the fish in a chemical bath to kill the parasites. The

chemical is cheap and most effective against the parasite at high doses, however, the fish are then more liable to suffer complications from the treatment and may die as a result. He therefore requires a suitable model with which to determine how he should be treating his fish.

Figure 4.16 presents a causal probabilistic network for the sick fish problem. It consists of six discrete variables. Of these, four, $C$, $D$, $S$ and $U$, are of finite range and the remaining two, $P$ and $R$, are of nonfinite range. We let $P$, the number of parasites on the fish before treatment, take a Poisson distribution. $D$, the chemical dose given to the fish, may take five levels ranging from a 20% dose to a 100% dose. $C$, the severity of the complications suffered by the fish takes three levels - none, minor and severe. $R$, the number of parasites remaining on the fish following chemical treatment, takes a Binomial distribution which may not be modelled by a finite range of probabilities since the original number of parasites, $P$, was not confined to a finite range. $S$, the survival of the fish, is binary - the fish either lives or dies. Similarly, $U$, whether the fish is useable as a prize 24 hours after treatment or not, takes a binary variable. The junction tree for the sick fish problem is given in Figure 4.17. Conditional probability tables for the model are presented in Table 4.22. There are two symbolic parameters required by the model, $p$ and $r$. These are the number of parasites on the fish before and after treatment, respectively. These parameters take non-negative integers.

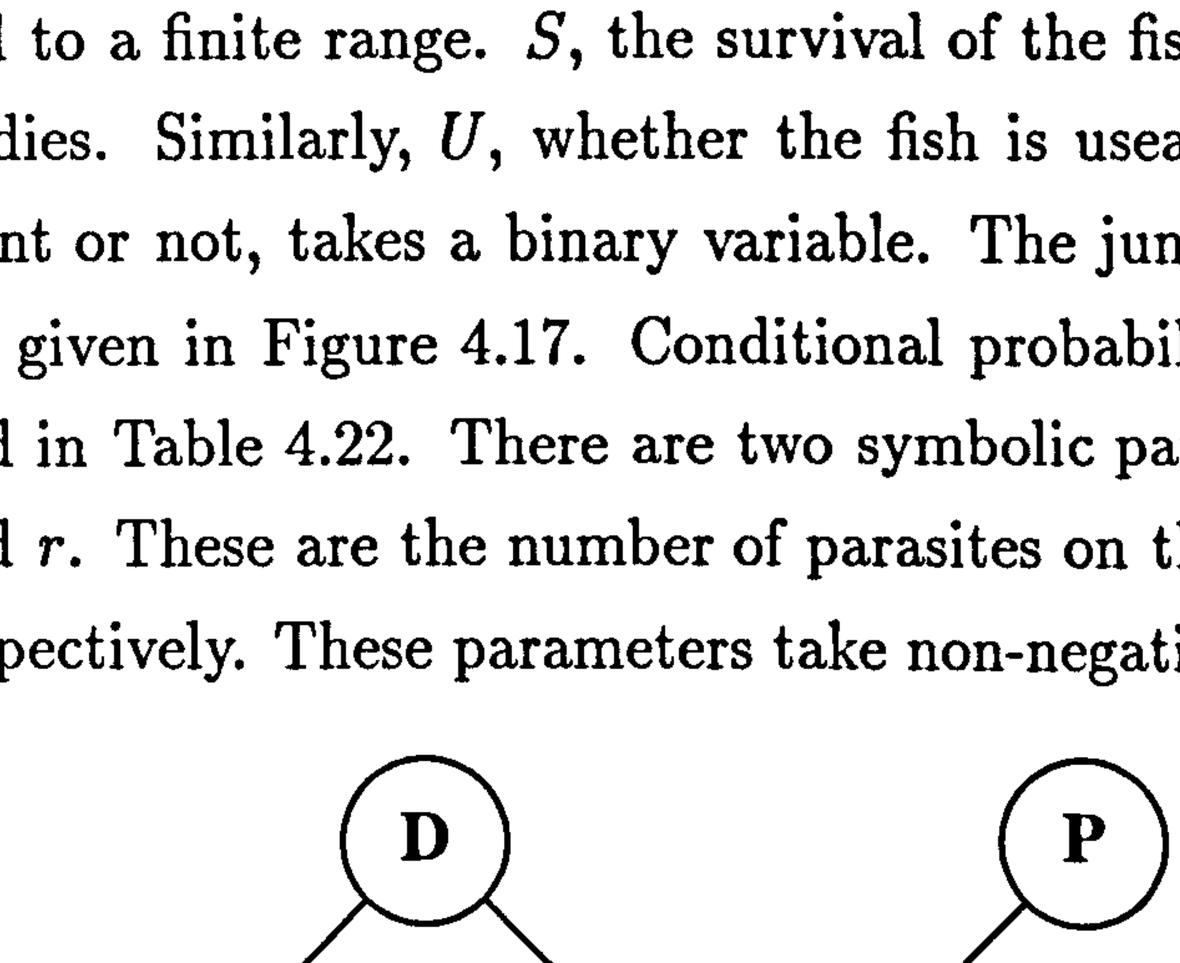



Figure 4.16: The causal probabilistic network illustrating the sick fish problem.

| $P$: # Parasites Before Treatment |
|---|
| $\frac{e^{-50}50^p}{p!}$ |

| $D$: Chemical Dose | | | | |
|---|---|---|---|---|
| 20% | 40% | 60% | 80% | 100% |
| $\frac{1}{10}$ | $\frac{1}{5}$ | $\frac{2}{5}$ | $\frac{1}{5}$ | $\frac{1}{10}$ |

| $C \mid D$ ($C$: Severity of Complications) | | | | | |
|---|---|---|---|---|---|
| | $D$ | | | | |
| $C$ | 20% | 40% | 60% | 80% | 100% |
| none | $\frac{4}{5}$ | $\frac{7}{10}$ | $\frac{3}{5}$ | $\frac{9}{20}$ | $\frac{1}{4}$ |
| minor | $\frac{3}{20}$ | $\frac{1}{5}$ | $\frac{1}{4}$ | $\frac{3}{10}$ | $\frac{7}{20}$ |
| severe | $\frac{1}{20}$ | $\frac{1}{10}$ | $\frac{3}{20}$ | $\frac{1}{4}$ | $\frac{2}{5}$ |

| $R \mid D, P$ ($R$: # Parasites After Treatment) | | | | |
|---|---|---|---|---|
| $D$ | | | | |
| 20% | 40% | 60% | 80% | 100% |
| $\binom{p}{r}\frac{7}{10}^r\frac{3}{10}^{p-r}$ | $\binom{p}{r}\frac{3}{5}^r\frac{2}{5}^{p-r}$ | $\binom{p}{r}\frac{1}{2}^p$ | $\binom{p}{r}\frac{2}{5}^r\frac{3}{5}^{p-r}$ | $\binom{p}{r}\frac{3}{10}^r\frac{7}{10}^{p-r}$ |

| $S \mid C, R$ ($S$: Survival) | | | |
|---|---|---|---|
| | $C$ | | |
| $S$ | none | minor | severe |
| lives | 1 | $\frac{99}{100}^r$ | $\frac{95}{100}^r$ |
| dies | 0 | $1-\frac{99}{100}^r$ | $1-\frac{95}{100}^r$ |

| $U \mid R, S$ ($U$:Useable as Prize) | | |
|---|---|---|
| | $S$ | |
| $U$ | lives | dies |
| useable | $\frac{98}{100}^r$ | 0 |
| useless | $1-\frac{98}{100}^r$ | 1 |

Table 4.22: The conditional probability tables of the variables in the sick fish example for $p = 0, 1, 2, \ldots$ and $r = 0, 1, \ldots, p$.


Figure 4.17: The junction tree illustrating the sick fish problem.

The sick fish problem may be programmed in *Mathematica* in a similar way to the chest clinic and genetics counselling problems. We will therefore not go into too much detail as to the data structures or basic code required. Additions to the code will be needed, however, to enable the modelling of the variables with *nonfinite* state space. There are two variables in the sick fish problem with nonfinite state space. One, $P$, has an infinite state space, the other, $R$, has state space the maximum dimension of which is dependent on $P$. Neither of these variables may be fully represented through a fixed set of cells in a potential table. The code must therefore be altered to allow for this. In practice we treat discrete variables with nonfinite state space similarly to the way we treated the continuous variables in the waste incinerator example. We represent them by symbolic variables and define their probability densities as expressions which are functions of these symbolic variables.

Arguably the hardest task for the programmer will be the definition of summation routines which will deal with the marginalisation of potentials with respect to discrete variables with nonfinite state space. These routines may be programmed similarly to the integration operator we introduced for continuous variables. We require summation rules based on the pattern matching of known symbolic forms and simplification rules which will keep the functions manageable. The exact range of functions required will depend on the problem at hand and we do not intend to present an exhaustive list of these, but rather just those needed by the sick fish problem. Symbolic languages, such as *Mathematica*, may have the appropriate simplification and marginalisation functions built-in as standard. Recall, for example, how we adopted a "black-box" approach to simplification in the genetics counselling problem. Providing, however, that the language has a flexible pattern-matching environment well-defined replacement rules can be constructed on an *ad-hoc* basis.

The sick fish problem has been carefully constructed to illustrate how different discrete functional forms may be recognised, manipulated and operated upon. Unlike the waste incinerator example in the previous section, it is not meant to

illustrate a family of distributions which, when combined, will form a joint distribution of a particular known form. There are therefore several functional forms which our code must be familiar with. Due to the construction of the problem all the required marginalisations may be solved exactly using algebraic or arithmetic methods. The summation routines we require are based upon Binomial and Poisson distributions. We shall list each set in turn together with the *Mathematica* code required to perform the summation. Note that the user defined exponential operator **exp** is used.

**Binomial Summations:**

**B1.** Binomial$(n, r/(r+s))$ for x $= (0, n)$.

$$\begin{aligned}\sum_{x=0}^{n} \frac{k r^x s^{n-x}}{x!(n-x)!} &= \frac{k(r+s)^n}{n!} \sum_{x=0}^{n} \binom{n}{x} \left(\frac{r}{r+s}\right)^x \left(\frac{s}{r+s}\right)^{n-x} \\ &= \frac{k(r+s)^n}{n!} \quad \text{if} \quad 0 \leq \frac{r}{r+s} \leq 1\end{aligned}$$

```
sum[((k_. * (r_^x_) * (s_^(n_ - x_)))/((x_!) * (n_ - x_)!)),{x_, 0, n_}] :=
    ((k * ((r + s)^n))/n!)/; (FreeQ[k, x]&&(0 <= (r/(r + s)) <= 1))
```

**B2.** Binomial$(n, 1/2)$ for x $= (0, n)$.

$$\begin{aligned}\sum_{x=0}^{n} \frac{k r^n}{x!(n-x)!} &= \frac{k(2r)^n}{n!} \sum_{x=0}^{n} \binom{n}{x} \frac{1}{2}^n \\ &= \frac{k(2r)^n}{n!}\end{aligned}$$

```
sum[((k_. * ((r_)^n_))/((x_!) * (n_ - x_)!)),{x_, 0, n_}] :=
                                    (k * ((2 * r)^n))/n!/; FreeQ[k, x]
```

**B3.** Binomial$(n, r/(1+r))$ for x $= (0, n)$.

$$\begin{aligned}\sum_{x=0}^{n} \frac{k r^x}{x!(n-x)!} &= k \sum_{x=0}^{n} \frac{\left((1+r)\left(\frac{r}{1+r}\right)\right)^x \left((1+r)\left(1-\frac{r}{1+r}\right)\right)^{n-x}}{x!(n-x)!} \\ &= \frac{k(1+r)^n}{n!} \sum_{x=0}^{n} \binom{n}{x} \left(\frac{r}{1+r}\right)^x \left(\frac{1}{1+r}\right)^{n-x} \\ &= \frac{k(1+r)^n}{n!} \quad \text{if} \quad 0 \leq \frac{r}{1+r} \leq 1\end{aligned}$$

```
sum[((k_. * ((r_)^x_))/((x_!) * (n_ - x_)!)),{x_, 0, n_}] :=
    ((k * ((r + 1)^n))/n!)/; (FreeQ[k, x]&&(0 <= (r/(r + 1)) < 1))
```

## Poisson Summations:

**P1.** Poisson($a$) for x $= (0, \infty)$.

$$\begin{aligned} \sum_{x=0}^{\infty} \frac{ka^x}{x!} &= ke^a \sum_{x=0}^{\infty} \frac{e^{-a}a^x}{x!} \\ &= ke^a \quad \text{if} \quad a > 0 \end{aligned}$$

```
sum[((k_. * a_^x_)/(x_!)),{x_, 0, Infinity}] :=
                                    k * exp[a]/; FreeQ[k, x]
```

**P2.** Poisson(1) for x $= (0, \infty)$.

$$\begin{aligned} \sum_{x=0}^{\infty} \frac{k}{x!} &= ke \sum_{x=0}^{\infty} \frac{e^{-1}}{x!} \\ &= ke \end{aligned}$$

```
sum[((k_.)/(x_!)),{x_, 0, Infinity}] :=
                                    k * exp[1]/; FreeQ[k, x]
```

**P3.** Poisson($a$) for x $= (s, \infty)$.

$$\begin{aligned} \sum_{x=s}^{\infty} \frac{ka^x}{(x-s)!} &= ka^s e^a \sum_{x=s}^{\infty} \frac{e^{-a}a^{x-s}}{(x-s)!} \\ &= ka^s e^a \quad \text{if} \quad a > 0 \end{aligned}$$

```
sum[k_. * (a_^x)/((x_ - s_)!),{x_, s_, Infinity}] :=
                                    k * a^s * exp[a]/; FreeQ[k, x]
```

**P4.** Poisson($ab$) for x $= (s, \infty)$.

$$\begin{aligned} \sum_{x=s}^{\infty} \frac{ka^x b^{x-s}}{(x-s)!} &= ka^s e^{ab} \sum_{x=s}^{\infty} \frac{e^{-ab}(ab)^{x-s}}{(x-s)!} \\ &= ka^s e^{ab} \quad \text{if} \quad ab > 0 \end{aligned}$$

```
sum[k_. * (a_^x_) * (b_^(x_ - s_))/((x_ - s_)!),{x_, s_, Infinity}] :=
                                    k * a^s * exp[a * b]/; FreeQ[k, x]
```

In order that we may keep the potentials functions manageable we employ the use of a simplification routine, or *simplifier*. The potential functions are simplified after each major operation in the passage of a flow (division, multiplication

and marginalisation). The simplifier is applied repeatedly until no further simplification may be carried out. The *Mathematica* routine `simp` will perform this task. The definition of `simp[{x_}]` ensures that the list structure of the potential tables is maintained. The command `(x1 === x2)` yields true if and only if the expression `x1` is identical to `x2` and thus controls the repeated application of the recursive simplifier `simple`.

```
simp[{x_}] :={simp[x]};
simp[x_] := Block[{x1, x2},
      x1 = x;
      x2 = simple[x1];
      While[Not[(x1 === x2)],
        x1 = x2;
        x2 = simple[x2];
      ];
      Return[x1];
]
simple[(a_+b_)/c_] := simple[a/c]+simple[b/c];
simple[(a_)/(b_)] := simple[a]/simple[b];
simple[(a_)+(b_)] := simple[a]+simple[b];
simple[(a_)^(x_)*(b_)^(x_)*(k_)] := simple[k*(a*b)^x];
simple[(a_)^(x_)*(b_)^(-x_)*(k_)] := simple[k*(a/b)^x];
simple[exp[a_]*exp[b_]*(k_)] := simple[k*exp[a+b]];
simple[k_*(a_+b_)] := simple[k*a]+simple[k*b];
simple[x_] := x;
```

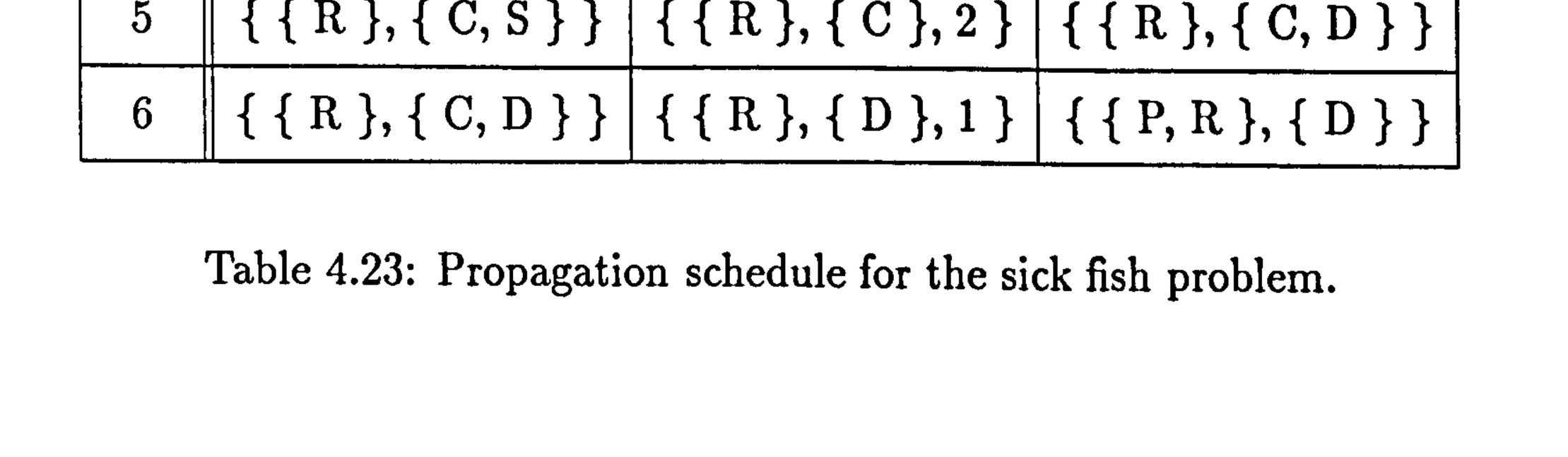
We assign the variables $D$, $P$, and $R$ to clique $\{D, P, R\}$, $C$ to $\{C, D, R\}$, $S$ to $\{C, R, S\}$, and $U$ to $\{R, S, U\}$. The propagation schedule is as given in Table 4.23. We define each clique as a list of two lists the former listing the symbolic variables in the clique, the latter gives an ordered list of the discrete variables of finite range in the clique. Their order in the list defines their order in the associated potential table. Each separator additionally has a third element - a unique number to allow for the possibility of separators with identical members. We assume that initialisation, propagation and evidence entry are applied in the usual way using the marginaliser and simplifier outlined here.

Under no evidence propagation requires the use of the symbolic summation rules **P3** and **P4** only. These are required for the first flow. All other flows require

simple summations. To determine the marginal distributions of all the variables we additionally need to use rules **B1**, **B2**, and **P1**. The marginal distributions for $C$, $D$, $S$, and $U$, the discrete variables of finite range, are purely numeric. The marginal distribution for $P$ is a Poisson distribution and the marginal distribution for $R$ is a Poisson mixture distribution. Plots of $P$ and $R$ for varying numbers of parasites are given in Figure 4.18. The marginal distribution of the variables is given in Table 4.24.

| No. | Source Clique | Separator | Sink clique |
|---|---|---|---|
| 1 | { { P, R }, { D } } | { { R }, { D }, 1 } | { { R }, { C, D } } |
| 2 | { { R }, { C, D } } | { { R }, { C }, 2 } | { { R }, { C, S } } |
| 3 | { { R }, { C, S } } | { { R }, { S }, 3 } | { { R }, { S, U } } |
| 4 | { { R, S }, { U } } | { { R }, { S }, 3 } | { { R }, { C, S } } |
| 5 | { { R }, { C, S } } | { { R }, { C }, 2 } | { { R }, { C, D } } |
| 6 | { { R }, { C, D } } | { { R }, { D }, 1 } | { { P, R }, { D } } |

Table 4.23: Propagation schedule for the sick fish problem.




$P$ : # Parasites Before Treatment $R$ : # Parasites After Treatment

Figure 4.18: The marginal distributions of $P$ and $R$ given no evidence.

| $P$: # Parasites Before Treatment |
|---|
| $\frac{50^p * exp[-50]}{p!}$ |

| $R$: # Parasites After Treatment |
|---|
| $\frac{35^r * exp[-35]}{10 * r!} + \frac{30^r * exp[-30]}{5 * r!} + \frac{2 * 25^r * exp[-25]}{5 * r!} + \frac{20^r * exp[-20]}{5 * r!} + \frac{15^r * exp[-15]}{10 * r!}$ |

| $C$: Severity of Complications | | |
|---|---|---|
| none | minor | severe |
| $\frac{23}{40}$ | $\frac{1}{4}$ | $\frac{7}{40}$ |

| $D$: Chemical Dose | | | | |
|---|---|---|---|---|
| 20% | 40% | 60% | 80% | 100% |
| $\frac{1}{10}$ | $\frac{1}{5}$ | $\frac{2}{5}$ | $\frac{1}{5}$ | $\frac{1}{10}$ |

| $S$: Survival | |
|---|---|
| lives | dies |
| $\frac{23}{40} + \frac{exp[\frac{-7}{4}]}{200} + \frac{exp[\frac{-3}{2}]}{50} + \frac{3 * exp[\frac{-5}{4}]}{50} + \frac{exp[-1]}{20} + \frac{exp[\frac{-3}{4}]}{25} + \frac{3 * exp[\frac{-7}{20}]}{200} + \frac{exp[\frac{-3}{10}]}{25} + \frac{exp[\frac{-1}{4}]}{10} + \frac{3 * exp[\frac{-1}{5}]}{50} + \frac{7 * exp[\frac{-3}{20}]}{200} =$ $= 0.832142$ | $\frac{17}{40} - \frac{exp[\frac{-7}{4}]}{200} - \frac{exp[\frac{-3}{2}]}{50} - \frac{3 * exp[\frac{-5}{4}]}{50} - \frac{exp[-1]}{20} - \frac{exp[\frac{-3}{4}]}{25} - \frac{3 * exp[\frac{-7}{20}]}{200} - \frac{exp[\frac{-3}{10}]}{25} - \frac{exp[\frac{-1}{4}]}{10} - \frac{3 * exp[\frac{-1}{5}]}{50} - \frac{7 * exp[\frac{-3}{20}]}{200} =$ $= 0.167858$ |

| $U$: Useable as Prize | |
|---|---|
| useable | useless |
| $\frac{exp[\frac{-483}{200}]}{200} + \frac{exp[\frac{-207}{100}]}{50} + \frac{3 * exp[\frac{-69}{40}]}{50} + \frac{exp[\frac{-69}{50}]}{20} + \frac{3 * exp[\frac{-1043}{1000}]}{200} + \frac{exp[\frac{-207}{200}]}{25} + \frac{exp[\frac{-447}{500}]}{25} + \frac{exp[\frac{-149}{200}]}{10} + \frac{2 * exp[\frac{-7}{10}]}{25} + \frac{7 * exp[\frac{-3}{5}]}{50} + \frac{3 * exp[\frac{-149}{250}]}{50} + \frac{6 * exp[\frac{-1}{2}]}{25} + \frac{7 * exp[\frac{-447}{1000}]}{200} + \frac{9 * exp[\frac{-2}{5}]}{100} + \frac{exp[\frac{-3}{10}]}{40} =$ $= 0.505991$ | $1 - \frac{exp[\frac{-483}{200}]}{200} - \frac{exp[\frac{-207}{100}]}{50} - \frac{3 * exp[\frac{-69}{40}]}{50} - \frac{exp[\frac{-69}{50}]}{20} - \frac{3 * exp[\frac{-1043}{1000}]}{200} - \frac{exp[\frac{-207}{200}]}{25} - \frac{exp[\frac{-447}{500}]}{25} - \frac{exp[\frac{-149}{200}]}{10} - \frac{2 * exp[\frac{-7}{10}]}{25} - \frac{7 * exp[\frac{-3}{5}]}{50} - \frac{3 * exp[\frac{-149}{250}]}{50} - \frac{6 * exp[\frac{-1}{2}]}{25} - \frac{7 * exp[\frac{-447}{1000}]}{200} - \frac{9 * exp[\frac{-2}{5}]}{100} - \frac{exp[\frac{-3}{10}]}{40} =$ $= 0.494009$ |

Table 4.24: The marginal distribution of the variables in the sick fish problem given no evidence.

The fairground worker is able to vary one variable only. This is $D$, the dose of chemical with which he treats a fish. In order to determine his optimal strategy, which maximises the probability of the fish being useable as a prize, we should therefore add evidence on $D$. Since the optimal strategy might also vary depending on the number of parasites on the fish before treatment we should also add evidence on $P$. The variable $P$ has infinite range and is represented as a symbolic variable. It would therefore not be appropriate to add numeric evidence on $P$ but rather symbolic evidence should be added. We therefore need to consider a total of five situations of evidence entry.

| $C$: Severity of Complications | | |
|---|---|---|
| none | minor | severe |
| $\frac{4}{5}$ | $\frac{3}{20}$ | $\frac{1}{20}$ |

| $R$: # Parasites After Treatment |
|---|
| $\frac{p!}{(p-r)!r!}\frac{3}{10}^{p-r}\frac{7}{10}^{r}$ |

| $S$: Survival | |
|---|---|
| lives | dies |
| $\frac{4}{5} + \frac{\frac{193}{200}^{p}}{20} + \frac{3\frac{993}{1000}^{p}}{20}$ | $\frac{1}{5} - \frac{\frac{193}{200}^{p}}{20} - \frac{3\frac{993}{1000}^{p}}{20}$ |

| $U$: Useable as Prize | |
|---|---|
| useable | useless |
| $\frac{\frac{9517}{10000}^{p}}{20} + \frac{3\frac{48957}{50000}^{p}}{20} + \frac{4\frac{493}{500}^{p}}{5}$ | $1 - \frac{\frac{9517}{10000}^{p}}{20} - \frac{3\frac{48957}{50000}^{p}}{20} - \frac{4\frac{493}{500}^{p}}{5}$ |

Table 4.25: The marginal distribution of the variables in the sick fish problem given the evidence that $D = 20\%$ and $P = p$.

Table 4.25, above, presents the marginal distributions of the variables in the sick fish problem given the evidence that $D = 20\%$ and $P = p$. This in fact turns out to be the optimal strategy regardless of the number of parasites on the fish before treatment. We may determine this by considering the graphs of the marginal probabilities $P(U = \text{useable})$ for $D = 20\%, 40\%, 60\%, 80\%$, and

100% and $P = p$ in Figure 4.19 or by considering the corresponding marginal probabilities given in Table 4.26.

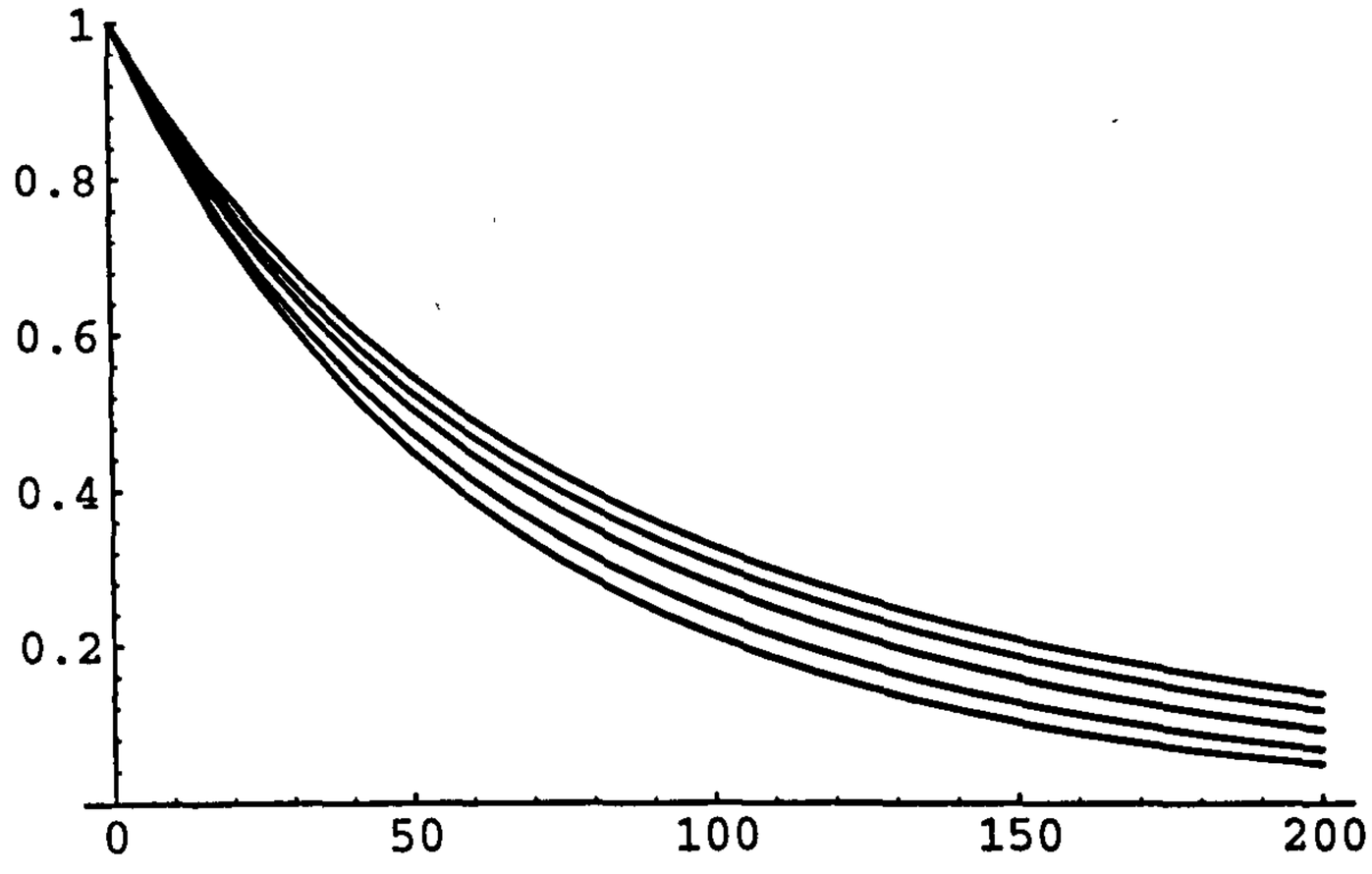



Figure 4.19: The probability that the fish is useable after treatment given numeric evidence on $D$ and symbolic evidence on $P$. The five lines from top to bottom represent the situations $D = 20\%$, $D = 40\%$, $D = 60\%$, $D = 80\%$, and $D = 100\%$ respectively for $P = p$.

| $U$: Useable as Prize | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $D$ | useable | | | | | | useless | | | | |
| 20% | $\frac{\frac{9517}{10000}p}{20}$ | + | $3\frac{\frac{48957}{50000}p}{20}$ | + | $4\frac{\frac{493}{500}p}{5}$ | | $1 - \frac{\frac{9517}{10000}p}{20} - 3\frac{\frac{48957}{50000}p}{20} - 4\frac{\frac{493}{500}p}{5}$ | | | | |
| 40% | $\frac{\frac{4793}{5000}p}{10}$ | + | $\frac{\frac{24553}{25000}p}{5}$ | + | $7\frac{\frac{247}{250}p}{10}$ | | $1 - \frac{\frac{4793}{5000}p}{10} - \frac{\frac{24553}{25000}p}{5} - 7\frac{\frac{247}{250}p}{10}$ | | | | |
| 60% | $\frac{3\frac{1931}{2000}p}{20}$ | + | $\frac{\frac{9851}{10000}p}{4}$ | + | $3\frac{\frac{99}{100}p}{5}$ | | $1 - \frac{3\frac{1931}{2000}p}{20} - \frac{\frac{9851}{10000}p}{4} - 3\frac{\frac{99}{100}p}{5}$ | | | | |
| 80% | $\frac{\frac{2431}{2500}p}{4}$ | + | $\frac{3\frac{12351}{12500}p}{10}$ | + | $9\frac{\frac{124}{125}p}{20}$ | | $1 - \frac{\frac{2431}{2500}p}{4} - \frac{3\frac{12351}{12500}p}{10} - 9\frac{\frac{124}{125}p}{20}$ | | | | |
| 100% | $\frac{2\frac{9793}{10000}p}{5}$ | + | $\frac{7\frac{49553}{50000}p}{20}$ | + | $\frac{\frac{497}{500}p}{4}$ | | $1 - \frac{2\frac{9793}{10000}p}{5} - \frac{7\frac{49553}{50000}p}{20} - \frac{\frac{497}{500}p}{4}$ | | | | |

Table 4.26: The marginal distribution of $U$ in the sick fish problem given the evidence that $D = 20\%, 40\%, 60\%, 80\%$, and $100\%$ for $P = p$.

## 4.15 Other Symbolic Techniques

In this chapter we have shown how global parameters, continuous variables, discrete variables with infinite range, and continuous evidence may be represented

by symbols which may then be manipulated using computer algebra. In this way we have shown how symbolic techniques may be used to extend the range of PESs which we may model using a computer. All these techniques have employed the use of symbolics on a *micro* level. In other words the symbolic techniques are, given the propagation algorithm, being applied at the level of the potential functions within each basic operator. We briefly mention here an alternative methodology which might be used in tandem with the other methods we have discussed. This is the use of symbolics at a *macro* level. Here symbolic techniques are used at the level of the basic operators and effectively determine the propagation algorithm.

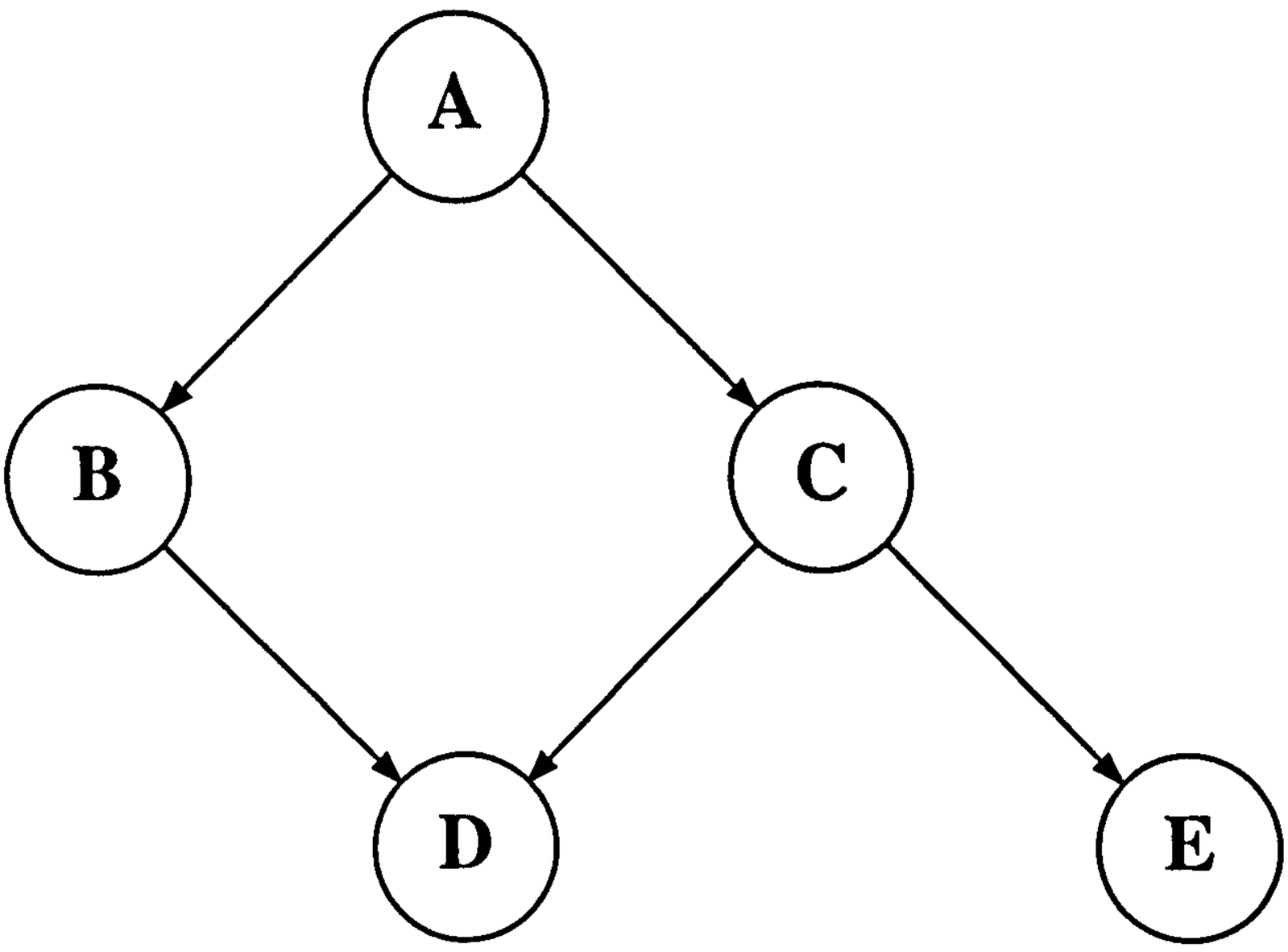



Figure 4.20: A simple belief network.

Symbolics at the macro level may be used to achieve a slightly different task to symbolics at the micro level but both adhear to the common goal of improving computational feasibility. At the macro level we are concerned with the manipulation of the basic operators and potential functions albeit in a somewhat abstract manner. Symbolic techniques may be applied to try to determine the optimal propagation technique which will derive a required result. Consider, for example, the discrete network of Figure 4.20 (from Li & D'Ambrosio, 1994) which is composed of binary variables only. Suppose we wish to determine the joint probability of variables $D$ and $E$, namely $p(d, e)$. One factoring is given in the formula:

$$p(d,e) = [\sum_a [\sum_b [\sum_c [p(e \mid c)p(d \mid b,c)]p(c \mid a)]p(b \mid a)]p(a)]$$

which requires 72 multiplications. Another factoring needs only 28 multiplications:

$$p(d,e) = [\sum_{c} [p(e \mid c)[\sum_{b} p(d \mid b,c)[\sum_{a} p(c \mid a)[p(b \mid a)p(a)]]]]]$$

It can thus be seen that different factoring can result in significantly different computational costs. Applying symbolic techniques to the first equation we could determine the second before having to translate the individual probabilities from abstract symbols to actual potential functions.

Symbolics at the macro level need not and, depending on the case, cannot be used in isolation. The symbolic probabilistic inference (SPI) approach (D'Ambrosio, 1994) applies symbolic techniques at a macro level in order to determine the optimal factoring of an individual marginal or conditional distribution. Given that optimal factoring any appropriate technique may then be used to determine the desired distribution. This may include the use of exact numeric techniques, approximation methods or even symbolic methods at the micro level.

# Chapter 5

# Spline Approximation Techniques

## 5.1 Introduction

In Chapter 4 we showed how symbolic techniques can enable us to model a particular form of mixed case. There is still a need, however, for methods which may be used to tackle the mixed distributional case in a more general fashion. The basic problem that arises in the modelling of such systems occurs during the marginalisation process where integrals which are either difficult or impossible to calculate exactly may be encountered. In this situation symbolic techniques can not be employed. Instead numeric methods, such as quadrature rules may be used but these do not possess the full generality that we desire and often extensive algebraic manipulation is required before they may be applied. Moreover, numeric integration techniques are prone to error and it may be difficult to spot such inaccuracies when they occur as part of a large automated system. Alternatively Monte-Carlo methods may be used (Dawid *et al.*, 1993) but these can fail completely in irregular problems such as in the case of multimodal distributions (Geweke, 1991). Monte-Carlo methods are usually defined as the computation of the expectation, under the posterior density, of some function of interest. We would like a method which is more flexible in its output and, in particular, will enable us to generate graphical results.

What we are seeking therefore is a method which will enable us to unite discrete and continuous methodology and will allow us to perform our marginalisations simply, and in the same way, for any given problem. We require that our method will facilitate graphical representations, as well as momental and distributional investigations. This chapter is concerned with establishing a basic framework for the solution of these problems through the use of splines. Splines provide a

compromise for the treatment of continuous variables in that the variables are constrained discretely to points on a multivariate grid yet they are interpolated continuously between these points. The marginalisation process is simplified since the true probability density function is approximated by polynomials which may be integrated exactly. Splines enable us to first investigate the approximation of each of the variables as a univariate problem. When we are satisfied with these approximations they may be used to model the multivariate problem. They also provide smooth graphs and are widely used in curve and surface fitting (Lancaster & Salkauskas, 1986). It is clear that since such methods are based on approximation techniques they may not be optimal when we are able to provide exact results by other methods. They should thus be used in tandem with exact methods using hybrid propagation algorithms (Dawid *et al.*, 1993) wherever possible.

## 5.2 Univariate Hermitian Interpolation

As an introduction to the use of splines let us consider how we may approximate some univariate function $g(z)$, say. Let us suppose we have a set of data for $g(z)$ which consists of a set of $M+1$ distinct points $z_0, z_1, \ldots, z_M$ and the corresponding ordinates $g_0, g_1, \ldots, g_M$ and slopes $g'_0, g'_1, \ldots, g'_M$. We shall term the $M+1$ points *knots*. We wish to find a function that will fit these data, and hence the true function $g$, exactly at the knots and interpolate for $g$ between them. A polynomial seems like a natural choice for our interpolating function as these are well understood and are easy to compute with. Since we have $2(M+1)$ conditions imposed by our data we could fit a unique polynomial of degree $2M+1$ to our data. Polynomials of high degree are, however, notorious in overfitting the data and we will obtain a highly oscillatory interpolant (Lancaster & Salkauskas, 1986). We will instead constrain ourselves to localising our fit to each of the $M$ intervals and thus obtain a piecewise polynomial interpolant $p(z)$. Considering a given interval $[z_m, z_{m+1}]$, for $m = 0, \ldots, M-1$, we will fit a polynomial $p_m(z)$ with the properties:

$$p_m(z_m) = g_m, \qquad p_m(z_{m+1}) = g_{m+1}, \qquad p'_m(z_m) = g'_m, \qquad p'_{m+1}(z_{m+1}) = g'_{m+1}$$

Since we have four constraints for each interval $[z_m, z_{m+1}]$ then $p_m(z)$ will be a uniquely determined cubic polynomial. Due to these constraints $p$, our interpolation for $g$, will be continuous at every point of $[z_0, z_M]$. We shall write this in the notation: $p \in C[z_0, z_M]$. Furthermore $p$ will have a continuous first derivative at

every point of $[z_0, z_M]$, written $p \in \mathcal{C}^1[z_0, z_M]$. Within each interval $[z_m, z_{m+1}]$, for $m = 0, \ldots, M-1$, our interpolant also has continuous second and third derivatives. Since $p_m(z)$ is a cubic, we may write the interpolant for a given interval $[z_m, z_{m+1}]$ as follows:

$$p_m(z) = \alpha(z - z_m)^3 + \beta(z - z_m)^2 + \gamma(z - z_m) + \delta \tag{5.1}$$

Differentiating Equation 5.1 with respect to $z$:

$$p'_m(z) = 3\alpha(z - z_m)^2 + 2\beta(z - z_m) + \gamma \tag{5.2}$$

Then at $z = z_m$ Equations 5.1 and 5.2 give us:

$$g_m = p_m(z_m) = \delta, \qquad g'_m = p'_m(z_m) = \gamma \tag{5.3}$$

Defining $h_m = z_{m+1} - z_m$ to be the interval width at $z = z_{m+1}$ we have:

$$\begin{aligned} g_{m+1} &= p_m(z_{m+1}) = \alpha h_m^3 + \beta h_m^2 + g'_m h_m + g_m \\ g'_{m+1} &= p'_m(z_{m+1}) = 3\alpha h_m^2 + 2\beta h_m + g'_m \end{aligned} \tag{5.4}$$

Solving Equations 5.4 with respect to $\alpha$ and $\beta$ gives:

$$\alpha = \frac{2(g_m - g_{m+1})}{h_m^3} + \frac{(g'_m + g'_{m+1})}{h_m^2}, \qquad \beta = \frac{3(g_{m+1} - g_m)}{h_m^2} - \frac{(2g'_m + g'_{m+1})}{h_m} \tag{5.5}$$

Hence from Equations 5.1, 5.3 and 5.5 our interpolant for $[z_m, z_{m+1}]$ is:

$$\begin{aligned} p_m(z) &= \left(\frac{2(g_m - g_{m+1})}{h_m^3} + \frac{(g'_m + g'_{m+1})}{h_m^2}\right)(z - z_m)^3 \\ &+ \left(\frac{3(g_{m+1} - g_m)}{h_m^2} - \frac{(2g'_m + g'_{m+1})}{h_m}\right)(z - z_m)^2 \\ &+ g'_m(z - z_m) + g_m \end{aligned} \tag{5.6}$$

Which we may write:

$$
\begin{aligned}
p_m(z) &= \left(\frac{2}{h_m^3}\left(z - z_m + \frac{h_m}{2}\right)(z - z_{m+1})^2\right) g_m \\
&+ \left(\frac{1}{h_m^2}(z - z_m)(z - z_{m+1})^2\right) g'_m \\
&+ \left(-\frac{2}{h_m^3}(z - z_m)^2\left(z - z_{m+1} - \frac{h_m}{2}\right)\right) g_{m+1} \\
&+ \left(\frac{1}{h_m^2}(z - z_m)^2(z - z_{m+1})\right) g'_{m+1} \qquad (5.7)
\end{aligned}
$$

This second representation of our interpolant is a summation of functions of $z$ multiplied by our data values - either an ordinate or a slope. We term each of these functions of $z$ a *cardinal function* and they provide a way in which we may characterise our interpolant of $g$ as a linear function of the ordinates and slopes. Let $p(z)$ be our piecewise polynomial interpolant for $g$ in the region $[z_0, z_M]$ then we may write $p(z)$ as follows:

$$
p(z) = \sum_{m=0}^{M} \Phi_m(z) g_m + \sum_{m=0}^{M} \Psi_m(z) g'_m \qquad (5.8)
$$

Where:

$$
\Phi_0(z) = \begin{cases}
\frac{2}{h_0^3}\left(z - z_0 + \frac{h_0}{2}\right)(z - z_1)^2 & : \quad z_0 \le z \le z_1 \\
0 & : \quad z_1 \le z \le z_M
\end{cases}
$$

$$
\Phi_m(z) = \begin{cases}
0 & : \quad z_0 \le z \le z_{m-1} \\
-\frac{2}{h_{m-1}^3}(z - z_{m-1})^2\left(z - z_m - \frac{h_{m-1}}{2}\right) & : \quad z_{m-1} \le z \le z_m \\
\frac{2}{h_m^3}\left(z - z_m + \frac{h_m}{2}\right)(z - z_{m+1})^2 & : \quad z_m \le z \le z_{m+1} \\
0 & : \quad z_{m+1} \le z \le z_M
\end{cases}
$$

$$
\text{for } m = 1, \ldots, M-1
$$

$$
\Phi_M(z) = \begin{cases}
0 & : \quad z_0 \le z \le z_{M-1} \\
-\frac{2}{h_{M-1}^3}(z - z_{M-1})^2\left(z - z_M - \frac{h_{M-1}}{2}\right) & : \quad z_{M-1} \le z \le z_M
\end{cases}
$$

$$\Psi_0(z) = \begin{cases} \frac{1}{h_0^2}(z-z_1)^2(z-z_0) & : \quad z_0 \le z \le z_1 \\ 0 & : \quad z_1 \le z \le z_M \end{cases}$$

$$\Psi_m(z) = \begin{cases} 0 & : \quad z_0 \le z \le z_{m-1} \\ \frac{1}{h_{m-1}^2}(z-z_{m-1})^2(z-z_m) & : \quad z_{m-1} \le z \le z_m \\ \frac{1}{h_m^2}(z-z_m)(z-z_{m+1})^2 & : \quad z_m \le z \le z_{m+1} \\ 0 & : \quad z_{m+1} \le z \le z_M \end{cases}$$

for $m = 1, \ldots, M-1$

$$\Psi_M(z) = \begin{cases} 0 & : \quad z_0 \le z \le z_{M-1} \\ \frac{1}{h_{M-1}^2}(z-z_{M-1})^2(z-z_M) & : \quad z_{M-1} \le z \le z_M \end{cases} \tag{5.9}$$

In Equation 5.9 the functions $\Phi$ and $\Psi$ are said to form a *cardinal basis* for the ordinates and slopes. Each *cardinal function* $\Phi$ solves the elementary interpolation problems:

$$\Phi_m(z_k) = \delta_{m,k}, \qquad m = 0,1,\ldots,M; \qquad k = 0,1,\ldots,M$$

and:

$$\Phi'_m(z_k) = 0, \qquad m = 0,1,\ldots,M; \qquad k = 0,1,\ldots,M$$

where $\delta_{m,k}$ is the Kronecker delta, taking the value 1 when $m = k$ and the value 0 when $m \neq k$. Similarly each *cardinal function* $\Psi$ solves the corresponding interpolation problems:

$$\Psi_m(z_k) = 0, \qquad m = 0,1,\ldots,M; \qquad k = 0,1,\ldots,M$$

and:

$$\Psi'_m(z_k) = \delta_{m,k}, \qquad m = 0,1,\ldots,M; \qquad k = 0,1,\ldots,M$$

The graphs of the cardinal functions $\Phi$ and $\Psi$ are given in Figures 5.1 and 5.2. It should be noted that $\Phi_0(z) \equiv 0$ when $z > z_1$, $\Phi_m(z) \equiv 0$ when $z < z_{m-1}$ or

$z > z_{m+1}$, and $\Phi_M(z) \equiv 0$ when $z < z_{M-1}$, similarly $\Psi_0(z) \equiv 0$ when $z > z_1$, $\Psi_m(z) \equiv 0$ when $z < z_{m-1}$ or $z > z_{m+1}$, and $\Psi_M(z) \equiv 0$ when $z < z_{M-1}$. The $\Phi_m$ and $\Psi_m$, for $m = 0, \ldots, M$, therefore have small support - i.e. they are only important in the interpolation process in the region spanned by at most three knots. The shape of the cardinal functions is determined by the knot widths $h_m$, for $m = 0, 1, \ldots, M$, only and thus, given a knot sequence, it is the ordinates and slopes which adjust the shape of the interpolating curve. The shape of the piecewise interpolant between any two knots is only dependent on the ordinates and slopes corresponding to those two knots.

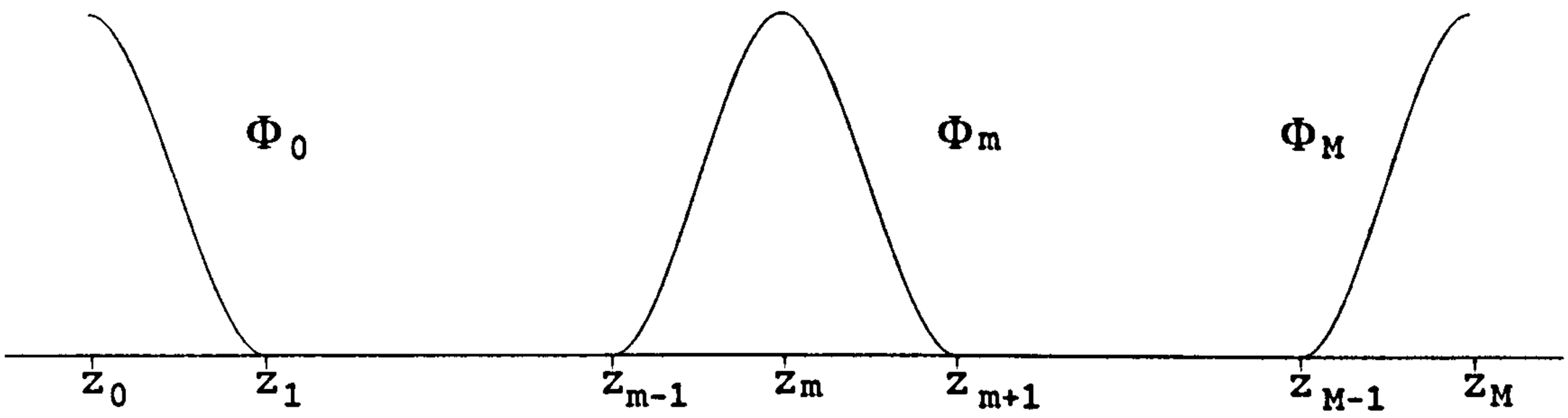



Figure 5.1: The Cardinal Functions $\Phi_m$.

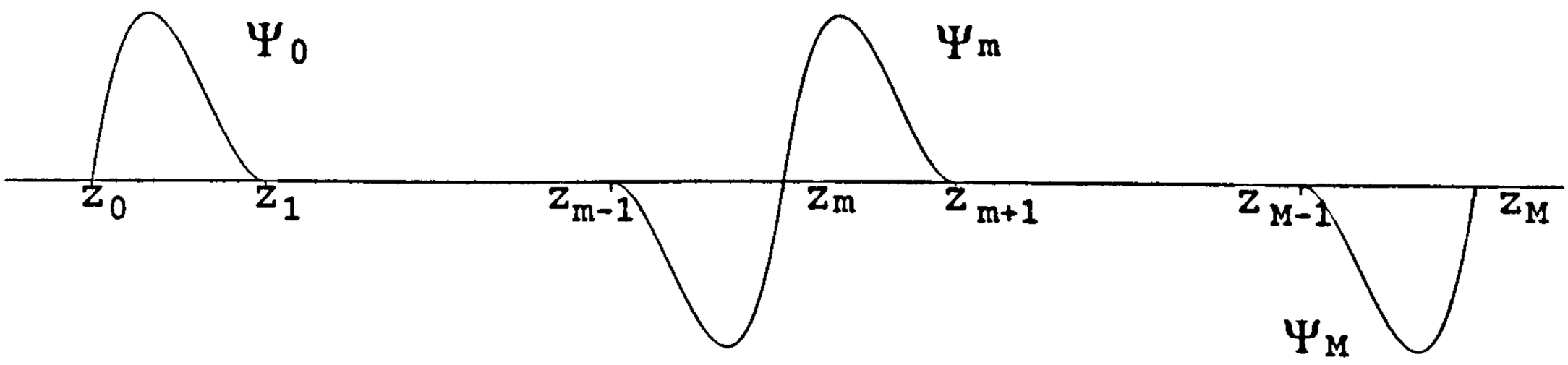



Figure 5.2: The Cardinal Functions $\Psi_m$.

We have a thus created a univariate scheme which forms an interpolating function for a known function given only its ordinates and slopes at a series of knots. This scheme is known as *Hermite interpolation*. The use of derivative data in the creation of our interpolating splines enables us to constrain the calculation of each spline to its corresponding interval only. Without such derivative data we would need to make assumptions of continuity in the first and second derivatives at the knots and subject our scheme to two further constraints in order to derive a unique interpolation given these constraints. The interpolant for an interval would then

be dependent on the entire data set, not just the data on the boundary of that interval.

## 5.3 Projectors

We now introduce the concept of a *projector* to enable us to manipulate our interpolation schemes more easily. Let $f$ be a function of class $\mathcal{L} = C[z_0, z_M]$, and $p$ be a uniquely defined interpolating function for $f$ of class $\mathcal{H}$ where $\mathcal{H}$ is a subspace of $\mathcal{L}$. Let $P$ be some function defined on $\mathcal{L}$ which has values in $\mathcal{H}$. Let us further assume that, for any $f \in \mathcal{L}$, $Pf = p$. $P$, therefore, may be said to *project* (or *map*) the functions of $\mathcal{L}$ onto $\mathcal{H}$, which we will call the *image* of $P$. Because $\mathcal{H}$ is contained in $\mathcal{L}$ we are able to apply the function $P$ once more to the function $Pf = p$. Writing $P^2$ for the composition of $P$ with itself, we have $P^2 f = Pp$. However, the definition of $P$ means that $Pp$ is the interpolant for $p$, which is just $p$. In other words $P$ is, in this case, just mapping $\mathcal{H}$ onto $\mathcal{H}$. Thus, for any $f \in \mathcal{L}$, $P^2 f = Pf$, which means that $P^2 = P$, and hence $P$ may be said to be *idempotent*.

A function $F$ on a vector space $\mathcal{L}$ with values in a vector space $\mathcal{H}$ is said to be *linear* if it is both *additive* and *homogeneous*, i.e.

$$F(z + w) = F(z) + F(w)$$

for all $z, w$ in $\mathcal{L}$, and

$$F(\alpha z) = \alpha F(z)$$

for any real $\alpha$ and all $z$ in $\mathcal{L}$. $F$ is then said to form a *linear transformation* from $\mathcal{L}$ into $\mathcal{H}$. We define a *projector* $P$ to be any transformation which is both linear and idempotent.

Consider the case of our univariate Hermitian interpolation as described in Section 5.2. We showed that given $M + 1$ distinct points $z_0, z_1, \ldots, z_M$ in $[z_0, z_M]$ and any function $f$, which, together with its first derivative, is well defined at these points, there is a unique piecewise cubic interpolant $p$ which takes the same function and derivative values as $f$ at $z_0, z_1, \ldots, z_M$. In the same way let us consider any function $f$ in $\mathcal{L} = C^1[z_0, z_M]$ and map it by Hermitian interpolation

onto a $p$ in $\mathcal{H} \subset \mathcal{L}$. This map determines the transformation $P$, say, and we may thus write $Pf = p$. Due to the uniqueness of the Hermitian interpolant, the function $P$ will simply reproduce functions $p$ in $\mathcal{H}$. Thus, if $Pf = p$ then $P^2 f = Pp = p$ so $P^2 f = Pf$ for any $f \in \mathcal{C}^1[z_0, z_M]$, hence $P$ is idempotent. $P$ is also linear, and so it is a projector of $\mathcal{L}$ onto $\mathcal{H}$. We showed in Equations 5.8 and 5.9 how $p$ may be written as a linear combination of the product of cardinal functions and the values of $f$ and $f'$ at the $M+1$ distinct points. We deduce that our interpolating projector $P$ may be expressed in terms of cardinal functions also:

$$p(z) = \sum_{l=0}^{2M+1} \Theta_l(z) f_l = Pf(z)$$

where $f_l = f_{2m}$ and $f_l = f_{2m+1}$ are the values $g_m$ and $g'_m$ from the original problem respectively, for $m = 0, 1, \ldots, M$. Similarly $\Theta_l = \Theta_{2m}$ and $\Theta_l = \Theta_{2m+1}$ are the cardinal functions $\Phi_m$ and $\Psi_m$ respectively, for $m = 0, 1, \ldots, M$. Thus the projector $P$ is the row vector of cardinal functions $[\Theta_0, \Theta_1, \ldots, \Theta_{2M+1}]$.

## 5.4 Multivariate Projection Methods

In order to link a particular knot with its function value or one of its derivatives at that point we introduce a structure consisting of both an abscissa and the order of the derivative connected with it. Such a structure is normally termed a *node* yet this may prove confusing since the term *node* has a different meaning in graph theory. We shall thus term it an *interpolation node*. The following definition is due to Lancaster & Salkauskas (1986):

**Definition 41** *If a function $f$ is known to belong to the class $C^R[z_0, z_M]$, and if the value of $f^{(r)}(z)$ is given for some $r$, $0 \leq r \leq R$, and $z \in [z_0, z_M]$, then the ordered pair $(z; r)$ is said to be an* ***interpolation node****. We refer to $f^{(r)}(z)$ as the* ***nodal value*** *of $f$ at this interpolation node.*

Now, as an introduction to the multivariate case, let us consider how we may combine two projectors. Suppose we have a projector associated with the interpolation of any function $f(z) \in \mathcal{L}_0$, where $\mathcal{L}_0 = C^R[z_0, z_M]$, with nodal values $f^{(r)}(z_m)$, $m = 0, 1, \ldots, M$, $0 \leq r \leq R$, given at a set of points $z_0, z_1, \ldots, z_M$ on a line. Further suppose that our interpolation scheme has been well-defined through a set of cardinal functions $\Theta_0^{(r)}(z), \Theta_1^{(r)}(z), \ldots, \Theta_M^{(r)}(z)$, for $r = 0, 1, \ldots, R$. Then for a particular function $f(z)$, say, there is a unique interpolant in the set $\mathcal{H}_0 \subset \mathcal{L}_0$

of linear combinations of the cardinal functions $\Theta_0^{(r)}(z), \Theta_1^{(r)}(z), \ldots, \Theta_M^{(r)}(z)$, for $r = 0, 1, \ldots, R$, given by:

$$(P_0 f)(z) = \sum_{r=0}^{R} \sum_{m=0}^{M} \Theta_m^{(r)}(z) f^{(r)}(z_m) \tag{5.10}$$

Here $P_0$ is the projector which maps functions in $\mathcal{L}_0$ to functions in $\mathcal{H}_0$. Let us now consider a projector $P$, formed as an extension of our definition of the projector $P_0$ in Equation 5.10, which applies to functions $f(z, w)$ which are at least continuous on $[z_0, z_M] \times [w_0, w_T]$. We may then write:

$$(Pf)(z, w) = \sum_{r=0}^{R} \sum_{m=0}^{M} \Theta_m^{(r)}(z) f^{(r)}(z_m, w) \tag{5.11}$$

$P$ is now a projector on horizontal lines in the $z$-plane which are determined by fixing the values of $w$. The image of $P$ consists of functions that, for each fixed value of $w$, are in $\mathcal{H}_0$, but for each fixed value of $z$ are merely functions continuous in $w$ on $[w_0, w_T]$. We may extend our definition of the projector $P$ in Equation 5.11 to higher dimensional systems by including more variables as "sleeping partners" in the functions $f$.

Now let us consider an interpolation scheme for $f(z, w)$ with respect to the $w$-variable which is determined by cardinal functions $\Psi_0^{(s)}(w), \Psi_1^{(s)}(w), \ldots, \Psi_T^{(s)}(w)$, for $s = 0, 1, \ldots, S$. We assume here that $f$ is continuous in $w$ up to and including the $S$th-derivative and we have nodal values given at a set of points $w_0, w_1, \ldots, w_T$ on a line. We may thus construct a projector $Q$ for $f$ in a similar fashion such that:

$$(Qf)(z, w) = \sum_{s=0}^{S} \sum_{t=0}^{T} \Psi_t^{(s)}(w) f^{(s)}(z, w_t) \tag{5.12}$$

$Q$ is now a projector on horizontal lines in the $w$-plane which are determined by fixing the values of $z$. The image of $Q$ consists of functions that, for each fixed value of $z$, are in $\mathcal{H}_1$, say, but for each fixed value of $w$ are merely functions continuous in $z$ on $[z_0, z_M]$.

We will now consider what we obtain when we apply the projector $P$ in Equation 5.11 to the definition of $Qf$ in Equation 5.12, which is:

$$(PQf)(z,w) = \sum_{r=0}^{R}\sum_{m=0}^{M}\Theta_m^{(r)}(z)\sum_{s=0}^{S}\sum_{t=0}^{T}\Psi_t^{(s)}(w)f^{(r,s)}(z_m,w_t)$$

$$= \sum_{r=0}^{R}\sum_{s=0}^{S}\sum_{m=0}^{M}\sum_{t=0}^{T}\Theta_m^{(r)}(z)\Psi_t^{(s)}(w)f^{(r,s)}(z_m,w_t) \qquad (5.13)$$

where $f^{(r,s)}(z_m,w_t)$ is taken to be the nodal value at an interpolation node obtained by combining the interpolation node $(z_m;r)$ in the $z$-direction with the interpolation node $(w_t;s)$ in the $w$-direction. We shall write such an interpolation node as the 4-tuple $(z_m,w_t;r,s)$ and it is clear that its nodal value must be:

$$f^{(r,s)}(z_m,w_t) = \frac{\delta^{r+s}f(z_m,w_t)}{\delta z^r \delta w^s} \qquad (5.14)$$

Similarly, applying the projector $Q$ in Equation 5.12 to $Pf$, as defined in Equation 5.11, we obtain the expression:

$$(QPf)(z,w) = \sum_{s=0}^{S}\sum_{t=0}^{T}\Psi_t^{(s)}(w)\sum_{r=0}^{R}\sum_{m=0}^{M}\Theta_m^{(r)}(z)f^{(r,s)}(z_m,w_t)$$

$$= \sum_{r=0}^{R}\sum_{s=0}^{S}\sum_{m=0}^{M}\sum_{t=0}^{T}\Theta_m^{(r)}(z)\Psi_t^{(s)}(w)f^{(r,s)}(z_m,w_t) \qquad (5.15)$$

Thus from Equations 5.13 and 5.15 we see that $PQ = QP$ and hence the projectors $P$ and $Q$ are *commutative*. Due to this principle:

$$(PQ)^2 = P(QP)Q = P(PQ)Q = P^2Q^2 = PQ$$

and so $PQ$ must be a projector also. Now considering the surface interpolant $PQf = QPf$ we see that this interpolant corresponds with the underlying function $f$ in terms of its nodal values exactly at the interpolation nodes. Moreover, suppose we were to fix $z$ at one of the points $z_0, z_1, \ldots, z_M$ then our surface interpolant $PQf$ would correspond with our curve interpolant $Q$ at the points $w_0, w_1, \ldots, w_T$ only. Similarly fixing $w$ to be one of the points $w_0, w_1, \ldots, w_T$ then $PQf$ and $P$ would correspond at the points $z_0, z_1, \ldots, z_M$ only. The projector $PQ$ is known as a *tensor product interpolant* (see, for example, Hall, 1979) and it is easy to see how such a projector may be extended to fit higher dimensional problems. The advantage of such a projector is that it allows us to combine univariate interpolatory schemes

and apply them to the multivariate case. The disadvantage of the scheme is that the univariate and multivariate interpolatory schemes only equate at the vertices of the lattice on which the interpolation is based.

Let us now consider a surface interpolation scheme formed additively by combining the projectors $P$, $Q$ and $PQ$ as follows:

$$Bf(z,w) = (Pf)(z,w) + (Qf)(z,w) - (PQf)(z,w) \tag{5.16}$$

It can be seen from Equation 5.16 that $B$ is just the map $P+Q-PQ$ of $f$, and since:

$$\begin{aligned}
B^2 &= (P+Q-PQ)^2 \\
&= P^2 + PQ - P^2Q + QP + Q^2 - QPQ - PQP - PQ^2 + PQPQ \\
&= P + PQ - PQ + PQ + Q - PQ - PQ - PQ + PQ \\
&= P + Q - PQ \\
&= B
\end{aligned}$$

then $B$ must be a projector also. Suppose we were to fix $z$ at some point $z_u$ where $z_u \in \{z_0, z_1, \ldots, z_M\}$, then considering our interpolant $Bf$ fixed at this $z$ we have:

$$\begin{aligned}
Bf(z_u,w) &= (Pf)(z_u,w) + (Qf)(z_u,w) - (PQf)(z_u,w) \\
&= \sum_{r=0}^{R}\sum_{m=0}^{M} \Theta_m^{(r)}(z_u) f^{(r)}(z_m,w) + \sum_{s=0}^{S}\sum_{t=0}^{T} \Psi_t^{(s)}(w) f^{(s)}(z_u,w_t) \\
&\quad - \sum_{r=0}^{R}\sum_{s=0}^{S}\sum_{m=0}^{M}\sum_{t=0}^{T} \Theta_m^{(r)}(z_u)\,\Psi_t^{(s)}(w) f^{(r,s)}(z_m,w_t) \\
&= f(z_u,w) + \sum_{s=0}^{S}\sum_{t=0}^{T} \Psi_t^{(s)}(w) f^{(s)}(z_u,w_t) - f(z_u,w) \\
&= \sum_{s=0}^{S}\sum_{t=0}^{T} \Psi_t^{(s)}(w) f^{(s)}(z_u,w_t) \\
&= (Qf)(z_u,w)
\end{aligned} \tag{5.17}$$

So, from Equation 5.17, our bivariate interpolant $Bf$ equates to our univariate interpolant $Qf$ at $z = z_u$. In other words the projectors $B$ and $Q$ correspond in their interpolation of $f$ along a series of parallel lines in the $w$-direction. Similarly

considering our interpolant $Bf$ for fixed $w = w_v$, $w_v \in \{w_0, w_1, \ldots, w_T\}$ we find that:

$$Bf(z, w_v) = (Pf)(z, w_v)$$

Thus $Bf$ equates to $Pf$ at $w = w_v$, and $B$ and $P$ correspond in their interpolation of $f$ along a series of parallel lines in the $z$-direction. We term $B$ a *blended interpolant*. Like a tensor product interpolant a blended interpolant allows us to combine univariate interpolatory schemes and apply them to the multivariate case. Unlike a tensor product interpolant, however, the blended interpolant allows the univariate and multivariate interpolatory schemes to equate not only at the vertices of the lattice on which the interpolation is based, but also along the lines of this lattice. We term such an interpolant *transfinite* for this reason.

The way in which the univariate projectors $P$ and $Q$ (and hence $PQ$ ) were combined to form the projector $B$ is termed a *Boolean sum* which we will denote $P \oplus Q$ and define:

$$P \oplus Q = P + Q - PQ$$

Extending our theory to the case where we have $N$ univariate projectors $P_1, \ldots, P_N$ each of which only interpolates a function $f$ in terms of a single variable $Z_1, \ldots, Z_N$ respectively, we can see that the blended interpolant we require is simply their Boolean sum $B$ where:

$$B = P_1 \oplus P_2 \oplus \ldots \oplus P_N$$

For example in the case $N = 3$, based on three projectors $P$, $Q$, and $R$ we will require the projector $B$ where:

$$\begin{aligned} P \oplus Q \oplus R &= (P + Q - PQ) \oplus R \\ &= (P + Q - PQ) + R - (P + Q - PQ)R \\ &= P + Q + R - PQ - PR - QR + PQR \end{aligned}$$

The blended interpolation method may be considered to be superior to the tensor product scheme since it provides a transfinite interpolation. Despite this apparent superiority it is, with the possible exception of the bivariate case, difficult to envisage a general situation whereby blended splines will prove to be an appropriate

choice of methodology. The reason for this stems from the fact that the blended interpolation of an $N$-dimensional surface will still require the handling of nodal values which are themselves surfaces of up to $(N-1)$ dimensions. For our purposes this would demand that all functions describing such surfaces may be integrated to closed form wherever necessary. Of course, this requires a robust integration procedure which, as stated in the introduction to this chapter, is something we can not necessarily rely on. In addition, robust symbolic methods would, in general, need to be employed to handle functions of an arbitrary functional form.

In contrast, if tensor product interpolation were to be applied to all the variables in a function requiring interpolation then the nodal values thus formed would be purely numeric in nature. This approach removes the need for symbolic methods and confines the integration problem to the integration of the cardinal functions. Since the cardinal functions are multiplicatively related polynomials, each of which is a function of only one underlying variable, integration will not be problematic.

## 5.5 Cubic Spline Interpolation of a Potential Function

Suppose we have a collection of continuous random variables $Z = (Z_1, \ldots, Z_N)$. Let us consider how we may interpolate some non-negative function $f$, say, defined on these variables where $f$ may possibly, though not necessarily, be a probability density function. We shall term $f$ a *potential function*. Since, we will be considering multi-dimensional systems, our formulae are likely to be too unwieldy to express in full. We shall use the following notation to abbreviate our expressions:

$$f(z) = f(z_1, z_2, \ldots, z_N)$$

$$f^*(z_{n,m_n}) = f(z_1, \ldots, z_{n-1}, z_{n,m_n}, z_{n+1}, \ldots, z_N)$$

Thus we understand $f(z)$ to refer to a potential function defined on a collection of continuous random variables $Z = (Z_1, \ldots, Z_N)$ and $f(z)$ is an $N$-dimensional surface. $f^*(z_{n,m_n})$ refers to a potential function defined on a collection of $(N-1)$ continuous random variables $Z \backslash Z_n = (Z_1, \ldots, Z_{n-1}, Z_{n+1}, \ldots, Z_N)$ and $f^*(z_{n,m_n})$ is an $(N-1)$-dimensional surface.

We will assume that in approximating $f$ we may restrict ourselves to considering some domain $\mathcal{R} \subset \mathbb{R}^N$, such that outside this domain $f$ is zero everywhere. We will then term $f$ to be *flat* outside $\mathcal{R}$ and we shall let $\mathcal{R}$ denote the following domain:

$$\mathcal{R} = \prod_{n=1}^{N} [z_{n,0}, z_{n,M_n}]$$

We will make the further assumption that $f$ is zero on the boundary of $\mathcal{R}$. Let us partition the domain $[z_{n,0}, z_{n,M_n}]$ of each variable $Z_n$, for $n = 1, \ldots, N$, into $M_n$ intervals by defining a knot sequence consisting of $M_n + 1$ knots which we will denote:

$$z_{n,0} < z_{n,1} < \ldots < z_{n,M_n}$$

We shall call $h_{n,m_n}$ the distance between two successive knots $z_{n,m_n}$ and $z_{n,m_{n+1}}$. If the knot sequence for each given variable $Z_n$, for $n = 1, \ldots, N$, is equally spaced then we may write $h_{n,m_n} \equiv h_n$, for all $n = 1, \ldots, N$. We have thus defined a lattice in $\mathbb{R}^N$ with a total of $M$ vertices where:

$$M = \prod_{n=1}^{N} (M_n + 1)$$

On each of these $M$ vertices we shall place $2^N$ interpolation nodes. Each interpolation node will be the $2N$-tuple:

$$(z_1, z_2, \ldots, z_N; \lambda_1, \lambda_2, \ldots, \lambda_N)$$

where $\lambda_n$ is either 0 or 1 for $n = 1, \ldots, N$. The nodal value corresponding to such an interpolation node will then be:

$$\frac{\delta^{\lambda_1+\lambda_2+\ldots+\lambda_N} f(z_1, z_2, \ldots, z_N)}{\delta z_1^{\lambda_1} \delta z_2^{\lambda_2} \ldots \delta z_N^{\lambda_N}} \tag{5.18}$$

Let us assume that we are able to determine the true nodal value of $f$ for each of these $(M \times 2^N)$ interpolation nodes. In particular since we have assumed $f$ to be flat on the boundary of $\mathcal{R}$ all nodal values on the boundary of $\mathcal{R}$ must be zero also. In other words we have:

$$\frac{\delta^{\lambda_1+\lambda_2+\ldots+\lambda_N} f(z_{n,0})}{\delta z_1^{\lambda_1} \delta z_2^{\lambda_2} \ldots \delta z_N^{\lambda_N}} = \frac{\delta^{\lambda_1+\lambda_2+\ldots+\lambda_N} f(z_{n,M_n})}{\delta z_1^{\lambda_1} \delta z_2^{\lambda_2} \ldots \delta z_N^{\lambda_N}} \equiv 0 \tag{5.19}$$

for all $\lambda_j = 0$ or 1, $n = 1, \ldots, N$, and $j = 1, \ldots, N$.

Now, let us consider each variable $Z_n$, for $n = 1, \ldots, N$, separately. We may define for each $Z_n$ a univariate projector $P_n$ which interpolates for $f$ in the direction of $Z_n$. We will use the projectors $P_n$, for $n = 1, 2, \ldots, N$, to build our interpolation scheme for $f$. By assumption, when all the other variables are fixed, we have data on both the value of $f$ and the first derivative of $f$ with respect to $z_n$ at each of the knots. We will therefore fit piecewise cubic splines to $f$ in the direction of $Z_n$ using univariate Hermitian interpolation. As we saw earlier in Section 5.2, such an interpolation scheme may be defined through the use of a set of cardinal functions which we will call $\Psi_{n,l_n}$, for $n = 1, \ldots, N$, and $l_n = 0, \ldots, 2M_n + 1$. We shall arrange these cardinal functions in such a way that we may express our univariate interpolation scheme in terms of a single summation comprising both function values and derivative data. Hence we will separately define $\Psi_{n,l_n}$ for even $l_n = 2m_n$ and odd $l_n = 2m_n+1$, for $l_n = 0, \ldots, 2M_n+1$ and $m_n = 0, \ldots, M_n$. When $l_n = 2m_n$ is even let $\Psi_{n,l_n}$ be the cardinal function which corresponds to the interpolation nodes $(z_1, z_2, \ldots, z_{n-1}, z_n, z_{n+1}, \ldots, z_N; \lambda_1, \lambda_2, \ldots, \lambda_{n-1}, 0, \lambda_{n+1}, \ldots, \lambda_N)$ and when $l_n = 2m_n + 1$ is odd let $\Psi_{n,l_n}$ be the cardinal function which corresponds to the interpolation nodes $(z_1, z_2, \ldots, z_{n-1}, z_n, z_{n+1}, \ldots, z_N; \lambda_1, \lambda_2, \ldots, \lambda_{n-1}, 1, \lambda_{n+1}, \ldots, \lambda_N)$. Our univariate interpolation for $f$ in the direction of $Z_n$, when all other variables are fixed, is analogous to Equations 5.8 and 5.9 and is thus:

$$(P_n f)(z) = \sum_{l_n=0}^{2M_n+1} \Psi_{n,l_n}(z_n) \psi(z_{n,l_n})$$

where:

$$\psi(z_{n,l_n}) = \psi(z_{n,2m_n}) = f^*(z_{n,m_n})$$

and:

$$\psi(z_{n,l_n}) = \psi(z_{n,2m_n+1}) = \frac{\delta f^*(z_{n,m_n})}{\delta z_n} \tag{5.20}$$

for $l_n = 0, \ldots, 2M_n + 1$ and $m_n = 0, 1, \ldots, M_n$ and the cardinal functions $\Psi$ are defined as follows:

$$\Psi_{n,0}(z_n) = \begin{cases} \frac{2}{h_{n,0}^3}\left(z_n - z_{n,0} + \frac{h_{n,0}}{2}\right)(z_n - z_{n,1})^2 & : \quad z_{n,0} \leq z_n \leq z_{n,1} \\ 0 & : \quad z_{n,1} \leq z_n \leq z_{n,M_n} \end{cases}$$

$$\Psi_{n,2m_n}(z_n) = \begin{cases} 0 & : \ z_{n,0} \le z_n \le z_{n,m_n-1} \\ \frac{-2}{h^3_{n,m_n-1}}(z_n - z_{n,m_n-1})^2\left(z_n - z_{n,m_n} - \frac{h_{n,m_n-1}}{2}\right) & : \ z_{n,m_n-1} \le z_n \le z_{n,m_n} \\ \frac{2}{h^3_{n,m_n}}\left(z_n - z_{n,m_n} + \frac{h_{n,m_n}}{2}\right)(z_n - z_{n,m_n+1})^2 & : \ z_{n,m_n} \le z_n \le z_{n,m_n+1} \\ 0 & : \ z_{n,m_n+1} \le z_n \le z_{n,M_n} \end{cases}$$

for $m_n = 1, \ldots, M_n - 1$

$$\Psi_{n,2M_n}(z_n) = \begin{cases} 0 & : \ z_{n,0} \le z_n \le z_{n,M_n-1} \\ \frac{-2}{h^3_{n,M_n-1}}(z_n - z_{n,M_n-1})^2\left(z_n - z_{n,M_n} - \frac{h_{n,M_n-1}}{2}\right) & : \ z_{n,M_n-1} \le z_n \le z_{n,M_n} \end{cases}$$

$$\Psi_{n,1}(z_n) = \begin{cases} \frac{1}{h^2_{n,0}}(z_n - z_{n,1})^2(z_n - z_{n,0}) & : \ z_{n,0} \le z_n \le z_{n,1} \\ 0 & : \ z_{n,1} \le z_n \le z_{n,M_n} \end{cases}$$

$$\Psi_{n,2m_n+1}(z_n) = \begin{cases} 0 & : \ z_{n,0} \le z_n \le z_{n,m_n-1} \\ \frac{1}{h^2_{n,m_n-1}}(z_n - z_{n,m_n-1})^2(z_n - z_{n,m_n}) & : \ z_{n,m_n-1} \le z_n \le z_{n,m_n} \\ \frac{1}{h^2_{n,m_n}}(z_n - z_{n,m_n})(z_n - z_{n,m_n+1})^2 & : \ z_{n,m_n} \le z_n \le z_{n,m_n+1} \\ 0 & : \ z_{n,m_n+1} \le z_n \le z_{n,M_n} \end{cases}$$

for $m_n = 1, \ldots, M_n - 1$

$$\Psi_{n,2M_n+1}(z_n) = \begin{cases} 0 & : \ z_{n,0} \le z_n \le z_{n,M_n-1} \\ \frac{1}{h^2_{n,M_n-1}}(z_n - z_{n,M_n-1})^2(z_n - z_{n,M_n}) & : \ z_{n,M_n-1} \le z_n \le z_{n,M_n} \end{cases}$$

(5.21)

It is clear from our scheme that $f^*(z_{n,m_n})$ and $\frac{\delta f^*(z_{n,m_n})}{\delta z_n}$ will only be well defined at the vertices of the lattice.

## 5.6 Tensor Product Interpolation

Let us consider how we might form the tensor product interpolant of a potential function $f$ for our $N$-dimensional scheme. This will be defined as:

$$Pf\left(z_1, z_2, \ldots, z_N\right) = \sum_{l_1=0}^{2M_1+1} \sum_{l_2=0}^{2M_2+1} \cdots \sum_{l_N=0}^{2M_N+1} v_{l_1,l_2,\ldots,l_N} \Psi_{1,l_1}\left(z_1\right) \Psi_{2,l_2}\left(z_2\right) \ldots \Psi_{N,l_N}\left(z_N\right) \tag{5.22}$$

where $v_{l_1,l_2,\ldots,l_N}$ denotes the nodal value of an interpolation node $(z_{1,m_1}, z_{2,m_2}, \ldots, z_{N,m_N}; \lambda_1, \lambda_2, \ldots, \lambda_N)$ where, for every $n = 1, 2, \ldots, N$, if $l_n$ is even then $m_n = \frac{l_n}{2}$ and $\lambda_n = 0$ and if $l_n$ is odd then $m_n = \frac{(l_n-1)}{2}$ and $\lambda_n = 1$. Thus:

$$\begin{aligned} v_{l_1,l_2,\ldots,l_N} &= f^{(\lambda_1,\lambda_2,\ldots,\lambda_N)}\left(z_{1,m_1}, z_{2,m_2}, \ldots, z_{N,m_N}\right) \\ &= \left. \frac{\delta^{\lambda_1+\lambda_2+\ldots+\lambda_N} f\left(z_1, z_2, \ldots, z_N\right)}{\delta z_1^{\lambda_1} \delta z_2^{\lambda_2} \ldots \delta z_N^{\lambda_N}} \right|_{z_1=z_{1,m_1}, z_2=z_{2,m_2}, \ldots, z_N=z_{N,m_N}} \end{aligned} \tag{5.23}$$

The cardinal functions $\Psi_{n,l_n}$ for odd $l_n = 2m_n + 1$ and even $l_n = 2m_n$ are defined as in Equations 5.21 for $m_n = 0, 1, \ldots, M_n$, $l_n = 0, 1, \ldots, 2M_n + 1$, and $n = 1, \ldots, N$.

We can see from Equations 5.22 and 5.23 that the value of $Pf$ corresponding to any point $z$, say, in the region $\mathcal{R}$ is defined exactly from the $(M \times 2N)$ nodal values $v$ and the $N$ cardinal functions $\Psi$.

One of the key operations required in a PES is that of marginalisation. We will now consider what we will obtain if we marginalise $Pf$ with respect to a single variable with equally spaced knot sequence.

**Theorem 36** *Let the knot sequence for variable $Z_j$, $j \in \{1, 2, \ldots, N\}$, be equally spaced then the function $Pf(z \backslash z_j)$ formed by the marginalisation of $Pf(z)$ with respect to $Z_j$ is:*

$$\begin{aligned} Pf\left(z \backslash z_j\right) &= \sum_{l_1=0}^{2M_1+1} \cdots \sum_{l_{j-1}=0}^{2M_{j-1}+1} \sum_{l_{j+1}=0}^{2M_{j+1}+1} \cdots \sum_{l_N=0}^{2M_N+1} v_{l_1,\ldots,l_{j-1},l_{j+1},\ldots,l_N} \\ &\times \Psi_{1,l_1}\left(z_1\right) \ldots \Psi_{j-1,l_{j-1}}\left(z_{j-1}\right) \Psi_{j+1,l_{j+1}}\left(z_{j+1}\right) \ldots \Psi_{N,l_N}\left(z_N\right) \end{aligned} \tag{5.24}$$

*where:*

$$v_{l_1,\ldots,l_{j-1},l_{j+1},\ldots,l_N} = (h_j)\left(\sum_{m_j=0}^{M_j} v^{(0)}_{l_1,\ldots,l_{j-1},m_j,l_{j+1},\ldots,l_N}\right) \tag{5.25}$$

*and:*

$$\begin{aligned} & v^{(0)}_{l_1,\ldots,l_{j-1},m_j,l_{j+1},\ldots,l_N} \\ &= f^{(\lambda_1,\ldots,\lambda_{j-1},0,\lambda_{j+1},\ldots,\lambda_N)}\left(z_{1,m_1},\ldots,z_{j-1,m_{j-1}},z_{j,m_j},z_{j+1,m_{j+1}},\ldots,z_{N,m_N}\right) \\ &= \left.\frac{\delta^{\lambda_1+\ldots+\lambda_{j-1}+\lambda_{j+1}+\ldots+\lambda_N} f\left(z_1,\ldots,z_{j-1},z_j,z_{j+1},\ldots,z_N\right)}{\delta z_1^{\lambda_1}\ldots\delta z_{j-1}^{\lambda_{j-1}}\delta z_{j+1}^{\lambda_{j+1}}\ldots\delta z_N^{\lambda_N}}\right|_{z_1=z_{1,m_1},\ldots,z_N=z_{N,m_N}} \end{aligned} \tag{5.26}$$

*The cardinal functions $\Psi_{n,l_n}$ in Equation 5.24, are defined as in Equations 5.21 for odd $l_n = 2m_n + 1$ and even $l_n = 2m_n$ where $l_n = 0,1,\ldots,2M_n+1$, $m_n = 0,1,\ldots,M_n$ and $n = 1,\ldots,j-1,j+1,\ldots,N$. Similarly in Equation 5.26, if $l_n = 2m_n+1$ is odd then $\lambda_n = 1$ and if $l_n = 2m_n$ is even then $\lambda_n = 0$ for $l_n = 0,1,\ldots,2M_n+1$, $m_n = 0,1,\ldots,M_n$ and $n = 1,\ldots,j-1,j+1,\ldots,N$.*

**Proof.** The marginalisation of $Pf(z)$ with respect to the variable $Z_j$ is accomplished by integrating $Pf(z)$ with respect to $Z_j$ over the region $[z_{j,0}, z_{j,M_j}]$. We may thus determine that:

$$\begin{aligned} Pf\left(z\backslash z_j\right) &= \underset{z_j=z_{j,M_j}}{Pf\left(z_1,\ldots,z_{j-1},z_{j+1},\ldots,z_N\right)} \\ &= \int_{z_j=z_{j,0}}^{z_j=z_{j,M_j}} Pf\left(z_1,\ldots,z_N\right)dz_j \\ &= \int_{z_j=z_{j,0}}^{z_j=z_{j,M_j}} \sum_{l_1=0}^{2M_1+1}\cdots\sum_{l_N=0}^{2M_N+1} v_{l_1,\ldots,l_N}\Psi_{1,l_1}\left(z_1\right)\ldots\Psi_{N,l_N}\left(z_N\right)dz_j \\ &= \sum_{l_1=0}^{2M_1+1}\cdots\sum_{l_{j-1}=0}^{2M_{j-1}+1}\sum_{l_{j+1}=0}^{2M_{j+1}+1}\cdots\sum_{l_N=0}^{2M_N+1}\Psi_{1,l_1}\left(z_1\right)\ldots\Psi_{j-1,l_{j-1}}\left(z_{j-1}\right) \\ &\times\ \Psi_{j+1,l_{j+1}}\left(z_{j+1}\right)\ldots\Psi_{N,l_N}\left(z_N\right)\sum_{l_j=0}^{2M_j+1} v_{l_1,\ldots,l_N}\int_{z_j=z_{j,0}}^{z_j=z_{j,M_j}}\Psi_{j,l_j}\left(z_j\right)dz_j \end{aligned} \tag{5.27}$$

Now, expanding the final summation in Equation 5.27 we may show that:

$$\sum_{l_j=0}^{2M_j+1} v_{l_1,\ldots,l_N} \int_{z_j=z_{j,0}}^{z_j=z_{j,M_j}} \Psi_{j,l_j}(z_j)\,dz_j = \sum_{m_j=0}^{M_j} v^{(0)}_{l_1,\ldots,l_{j-1},m_j,l_{j+1},\ldots,l_N} \int_{z_j=z_{j,0}}^{z_j=z_{j,M_j}} \Psi_{j,2m_j}(z_j)\,dz_j + \sum_{m_j=0}^{M_j} v^{(1)}_{l_1,\ldots,l_{j-1},m_j,l_{j+1},\ldots,l_N} \int_{z_j=z_{j,0}}^{z_j=z_{j,M_j}} \Psi_{j,2m_j+1}(z_j)\,dz_j \tag{5.28}$$

where $v^{(0)}_{l_1,\ldots,l_{j-1},m_j,l_{j+1},\ldots,l_N}$ is as given in Equation 5.26 and $v^{(1)}_{l_1,\ldots,l_{j-1},m_j,l_{j+1},\ldots,l_N}$ is defined as follows:

$$\begin{aligned} & v^{(1)}_{l_1,\ldots,l_{j-1},m_j,l_{j+1},\ldots,l_N} \\ &= f^{(\lambda_1,\ldots,\lambda_{j-1},1,\lambda_{j+1},\ldots,\lambda_N)}\left(z_{1,m_1},\ldots,z_{j-1,m_{j-1}},z_{j,m_j},z_{j+1,m_{j+1}},\ldots,z_{N,m_N}\right) \\ &= \left.\frac{\delta^{\lambda_1+\ldots+\lambda_{j-1}+1+\lambda_{j+1}+\ldots+\lambda_N} f(z_1,\ldots,z_{j-1},z_j,z_{j+1},\ldots,z_N)}{\delta z_1^{\lambda_1}\ldots\delta z_{j-1}^{\lambda_{j-1}}\delta z_j \delta z_{j+1}^{\lambda_{j+1}}\ldots\delta z_N^{\lambda_N}}\right|_{z_1=z_{1,m_1},\ldots,z_N=z_{N,m_N}} \end{aligned} \tag{5.29}$$

Under the assumption that the knot sequence for $Z_j$ is equally spaced the knot width $h_{j,m_j} = (z_{j,m_j+1} - z_{j,m_j}) = h_j$, for all $m_j = 0,1,\ldots,M_j$ and we may then show that:

$$\int_{z_j=z_{j,0}}^{z_j=z_{j,M_j}} \Psi_{j,2m_j}(z_j)\,dz_j = \begin{cases} \frac{h_{j,0}}{2} = \frac{h_j}{2} & : m_j = 0 \\ \frac{1}{2}\left(h_{j,m_j-1} + h_{j,m_j}\right) = h_j & : m_j = 1,\ldots,M_j-1 \\ \frac{h_{j,M_j-1}}{2} = \frac{h_j}{2} & : m_j = M_j \end{cases}$$

$$\int_{z_j=z_{j,0}}^{z_j=z_{j,M_j}} \Psi_{j,2m_j+1}(z_j)\,dz_j = \begin{cases} \frac{h^2_{j,0}}{12} = \frac{h_j^2}{12} & : m_j = 0 \\ \frac{1}{12}\left(h^2_{j,m_j-1} - h^2_{j,m_j}\right) = 0 & : m_j = 1,\ldots,M_j-1 \\ \frac{h^2_{j,M_j-1}}{12} = \frac{h_j^2}{12} & : m_j = M_j \end{cases} \tag{5.30}$$

Now, incorporating the results of Equation 5.30 into Equation 5.28 we find that:

$$\begin{aligned}
\sum_{l_j=0}^{2M_j+1} v_{l_1,\ldots,l_N} \int_{z_j=z_{j,0}}^{z_j=z_{j,M_j}} \Psi_{j,l_j}(z_j)\,dz_j &= \left(v^{(0)}_{l_1,\ldots,l_{j-1},0,l_{j+1},\ldots,l_N}\right)\left(\frac{h_j}{2}\right) \\
&+ \left(\sum_{m_j=1}^{M_j-1} v^{(0)}_{l_1,\ldots,l_{j-1},m_j,l_{j+1},\ldots,l_N}\right)(h_j) \\
&+ \left(v^{(0)}_{l_1,\ldots,l_{j-1},M_j,l_{j+1},\ldots,l_N}\right)\left(\frac{h_j}{2}\right) \\
&+ \left(v^{(1)}_{l_1,\ldots,l_{j-1},0,l_{j+1},\ldots,l_N}\right)\left(\frac{h_j^2}{12}\right) \\
&+ \left(\sum_{m_j=1}^{M_j-1} v^{(1)}_{l_1,\ldots,l_{j-1},m_j,l_{j+1},\ldots,l_N}\right)(0) \\
&+ \left(v^{(1)}_{l_1,\ldots,l_{j-1},M_j,l_{j+1},\ldots,l_N}\right)\left(\frac{h_j^2}{12}\right)
\end{aligned} \tag{5.31}$$

But using our assumptions of flatness on the boundary of the region $\mathcal{R}$ $v^{(0)}_{l_1,\ldots,l_{j-1},0,l_{j+1},\ldots,l_N}$, $v^{(0)}_{l_1,\ldots,l_{j-1},M_j,l_{j+1},\ldots,l_N}$, $v^{(1)}_{l_1,\ldots,l_{j-1},0,l_{j+1},\ldots,l_N}$, and $v^{(1)}_{l_1,\ldots,l_{j-1},M_j,l_{j+1},\ldots,l_N}$ are all zero so, cancelling terms, Equation 5.31 may be rewritten:

$$(h_j)\left(\sum_{m_j=1}^{M_j-1} v^{(0)}_{l_1,\ldots,l_{j-1},m_j,l_{j+1},\ldots,l_N}\right)$$

or, equivalently:

$$(h_j)\left(\sum_{m_j=0}^{M_j} v^{(0)}_{l_1,\ldots,l_{j-1},m_j,l_{j+1},\ldots,l_N}\right) \tag{5.32}$$

which is $v_{l_1,\ldots,l_{j-1},l_{j+1},\ldots,l_N}$ as defined in Equation 5.25. Hence Equation 5.27 becomes the expression given by Equation 5.24.

□

Interpreting Theorem 36 we may marginalise a spline interpolated potential $Pf(z)$ over any given variable $Z_j$, for $j = 1, 2, \ldots, N$, with equally spaced knots by first dropping the information on the nodal values which have been differentiated with respect to $Z_j$ and then multiplying the knot width $h_j$ for $Z_j$ by the sum of the nodal values which have not been differentiated with respect to $Z_j$. Following

this computational scheme in order to find the marginal distribution of a single variable $Z_n$, for $n = 1, 2, \ldots, N$, we may marginalise over all the other variables $Z \backslash Z_n$ in turn, and, assuming that these variables have equally spaced knots, we will then obtain:

$$Pf(z_n) = \sum_{l_n=0}^{2M_n+1} v_{l_n} \Psi_{n,l_n}(z_n) \tag{5.33}$$

where:

$$v_{l_n} = \left( \prod_{\substack{j=1 \\ j \neq n}}^{N} h_j \right) \left( \sum_{m_1=0}^{M_1} \cdots \sum_{m_{n-1}=0}^{M_{n-1}} \sum_{m_{n+1}=0}^{M_{n+1}} \cdots \sum_{m_N=0}^{M_N} \left. \frac{\delta^{\lambda_n} f(z_1, \ldots, z_N)}{\delta z_n^{\lambda_n}} \right|_{\substack{z_1 = z_{1,m_1}, \ldots, \\ z_N = z_{1,m_N}}} \right) \tag{5.34}$$

The cardinal functions $\Psi_{n,l_n}$, for odd $l_n = 2m_n + 1$ and even $l_n = 2m_n$ are defined as in Equations 5.21 for $l_n = 0, 1, \ldots, 2M_n + 1$ and $m_n = 0, 1, \ldots, M_n$. Similarly, if $l_n = 2m_n + 1$ is odd then $\lambda_n = 1$ and if $l_n = 2m_n$ is even then $\lambda_n = 0$ for $l_n = 0, 1, \ldots, 2M_n + 1$ and $m_n = 0, 1, \ldots, M_n$.

We can see from Equations 5.33 and 5.34 that the interpolant for the marginal function of $f$ defined on $Z_n$ alone, is formed from the cardinal functions defined by the univariate projector $P_n$, and a total of $2M$ nodal values. It is clear therefore that if we require only the marginal distributions of the variables $Z_n$, for $n = 1, 2, \ldots, N$, derived from $Pf(z)$ then only $(M \times (N + 1))$ nodal values are actually ever used. This represents a saving of $(M \times [2^N - (N + 1)])$ nodal values from the $(M \times 2^N)$ nodal values considered earlier. It should also be noted that our assumptions of equally spaced interpolation nodes and flatness not only simplify our calculations considerably, but also ensure that the order in which we do our marginalisations has no effect on the results of the marginalisation. This is obviously a desirable property since the order in which one integrates should not affect the results of the integration.

## 5.7 Building a Probabilistic Expert System

In this section we will describe how probabilistic expert systems incorporating the spline interpolation of potential functions may be built. We will demonstrate these techniques for mixed graphical association models in which the continuous

variables are of Conditional Gaussian type, the discrete variables are of exact numeric type and no continuous variable has a discrete descendant. The Waste Incinerator problem will be modelled in this way.

### 5.7.1 Basic Structure

We shall now describe the basic structure required to form a mixed PES comprising spline interpolated continuous variables.

Let $\mathcal{G} = (K, E)$ be an independence graph for a set of random variables $X = (X_1, X_2, \ldots, X_k)$ such that a vertex $a \in K$ corresponds to a random variable $X_a$. Let us partition $K$ into a set of discrete vertices $\Delta$ and a set of continuous vertices $\Omega$ such that $K = \Delta \cup \Omega$. Let $I_\delta$ for $\delta \in \Delta$ denote a particular discrete variable $X_\delta$ and let $Z_\omega$ for $\omega \in \Omega$ denote a particular continuous variable $X_\omega$. $I_\delta$ takes values in a discrete space $\mathcal{X}_\delta = \mathcal{I}_\delta$ and $Z_\omega$ takes real values in a continuous space $\mathcal{X}_\omega = I\!R$. For any $A \subseteq K$ we write $\mathcal{X}_A$ for the space $\times_{a \in A} \mathcal{X}_a$ and in particular:

$$\mathcal{X}_A = \times_{a \in A} \mathcal{X}_a = \mathcal{I}_A \times \mathcal{Z}_A = (\times_{\delta \in A \cap \Delta} \mathcal{X}_\delta) \times (\times_{\omega \in A \cap \Omega} \mathcal{X}_\omega)$$

We abbreviate $\mathcal{X}_K$ to $\mathcal{X}$. If $x = (x_a : a \in K)$ then we let $x_A = (x_a : a \in A)$. A typical component $x$ of the joint state space of discrete and continuous variables $\mathcal{X}$ may be written in terms of its discrete and continuous components thus:

$$x = (x_a)_{a \in K} = (i, z) = \left((i_\delta)_{\delta \in \Delta}, (z_\omega)_{\omega \in \Omega}\right)$$

We will similarly write a typical component $x_A$ of the state space $\mathcal{X}_A$ as:

$$x = (x_a)_{a \in A} = (i_A, z_A) = \left((i_\delta)_{\delta \in A \cap \Delta}, (z_\omega)_{\omega \in A \cap \Omega}\right)$$

Each continuous variable $Z_\omega$, for $\omega \in \Omega$, will be spline interpolated. We will therefore need to constrain each $Z_\omega$ to take real values in the continuous space $\mathcal{R}_\omega \subset \mathcal{X}_\omega$. We will let $\mathcal{R}_\omega = [z_{\omega,0}, z_{\omega,M_\omega}]$ where $z_{\omega,0}$ and $z_{\omega,M_\omega}$ are the first and last knots for the interpolation of $Z_\omega$ and we assume that there are a total of $M_\omega + 1$ knots used in the interpolation of $Z_\omega$.

Associated with $\mathcal{G}$ is a junction tree $\mathcal{T}$ which we will assume is connected. Let $\mathcal{T}$ have vertex-set $\mathcal{C}$ and edge-set $\mathcal{S}$. Associated with any $C \in \mathcal{C}$ is a subset of $K$, which we shall denote by $C$ also and term a clique. Similarly associated with any $S \in \mathcal{S}$, joining two cliques $C$ and $C'$ is a subset of $K$ which we will denote by $S$ also. We shall term $S$ a separator and $S = C \cap C'$.

### 5.7.2 Mixed Graphical Association Models

To demonstrate our techniques with a known setup let us assume that no continuous vertex $a \in \Omega$ may have a discrete descendant. Then the probability of a discrete variable $I_a$, for $a \in \Delta$, given the states of its parents, which by assumption are all discrete, may be described by a table of probabilities $p(i_a; i_{pa(a)})$. We assume that this table is of finite dimension and the probabilities are known.

For each continuous variable $Z_a$, for $a \in \Omega$, we assume that the conditional distribution of $Z_a$ given its parents is conditionally Gaussian thus:

$$\begin{aligned} Z_a \mid X_{pa(a)} &= f\left(z_a; i_{pa(a)}, z_{pa(a)}\right) \\ &\sim N\left(\alpha(i_{pa(a)}) + \beta(i_{pa(a)})^T z_{pa(a)}, \sigma^2(i_{pa(a)})\right) \end{aligned}$$

It should be noted that none of these assumptions will be necessary for the application of our spline interpolated techniques. The assumption that continuous vertices do not have discrete ancestors will simplify our description of the choice of knot sequences, however. Similarly the distributional assumptions will enable the description of the Waste Incinerator Example.

### 5.7.3 Forming Univariate Interpolants

Knot sequences must be determined for each continuous variable $Z_a$, for $a \in \Omega$, such that the knots for each $Z_a$ are equally spaced and envelop a region which is sufficiently large enough to ensure that the boundary conditions discussed in Section 5.5 apply to the conditional probability density functions $f(z_a; i_{pa(a)}, z_{pa(a)})$ for all levels of $I_{pa(a)} = i_{pa(a)}$. It is also important to make sure that, once the boundary knots have been fixed, the knot widths are small enough to give an adequate approximation to each of the conditional probability density functions $f(z_a; i_{pa(a)}, z_{pa(a)})$ for all levels of $I_{pa(a)} = i_{pa(a)}$. While increasing the number of knots will always provide a solution to this particular problem the number of knots required may in certain cases prove to be excessive and our method will fail to provide an attractive solution.

In determining the knot sequences we must consider a series of interconnected univariate interpolation problems which should be tackled in a methodical fashion. Since we have defined the conditional independence properties of our variables in

terms of a CPN and CPNs may be well-numbered, we may form a causal ordering of our continuous variables based on this well-numbering and use this ordering to determine the order in which we should form each knot sequence.

To determine the knot sequence for any particular variable $Z_a$, for $a \in \Omega$, we may proceed as follows, assuming that the knot sequences for $Z_{pa(a)}$ have already been defined according to our schedule:

**1.** Determine an appropriate boundary $[z_{a,0}(i_{pa(a)}, z_{pa(a)}), z_{a,M_a}(i_{pa(a)}, z_{pa(a)})]$ for $f(z_a; i_{pa(a)}, z_{pa(a)})$ for all levels of $I_{pa(a)} = i_{pa(a)}$.

**2.** Fix the boundary for $Z_a$ as:

$$[z_{a,0}, z_{a,M_a}] = [\min z_{a,0}(i_{pa(a)}, z_{pa(a)}), \max z_{a,M_a}(i_{pa(a)}, z_{pa(a)})]$$

**3.** Take any $f(z_a; i_{pa(a)}, z_{pa(a)})$ and increase the number of knots in blocks within the boundary $[z_{a,0}, z_{a,M_a}]$ until an adequate fit is obtained. The adequacy of such a fit may be determined by marginalising with respect to $Z_a$ and comparing the result with the true result of one. Since this marginalisation may be accomplished by summing the function values at each of the knots and multiplying the result by the knot width this is both easy and cheap to do. It should be noted that derivative values need not be calculated at this point.

**4.** Once a suitable knot sequence has been determined for one of the $f(z_a; i_{pa(a)}, z_{pa(a)})$ at $I_{pa(a)} = i_{pa(a)}$ see if it is adequate for the remainder. If not repeat steps **(3)** and **(4)** with different values of $I_{pa(a)}$ increasing the number of knots in blocks from the current value each time until an adequate approximation has been obtained for all the $f(z_a; i_{pa(a)}, z_{pa(a)})$. At this stage we may use a binary search to determine exactly where the optimal number of knots lies within the last block increase of knots. Since the marginalisation adequacy criterion for the knot sequence is a *necessary* condition for a good fit and not a *sufficient* one a less efficient technique may additionally be applied to check the fit. This could be a graphical technique in which the true and approximated distributions are graphed and compared by eye or an automated technique in which inter-knot points are used to check that the absolute differences between true and approximated function values at these points are within some tolerance range. Both these techniques require derivative values.

**5.** If steps **(1)** to **(4)** appear to be choosing too many knots either the desired adequacy of the approximation should be laxened or other methods should be used.

Having calculated the knot sequence for each of the continuous variables we must then calculate the nodal values of each of the continuous distributions. Such nodal values need only correspond to the functions and their first derivatives at the interpolation nodes.

### 5.7.4 Waste Incinerator Example

To illustrate our methodology let us consider how we might tackle Lauritzen's waste incinerator problem (Lauritzen, 1992). The CPN in Figure 5.3a shows the independence graph of the associated model with variables 'Burning Regime' ($B$), 'Filter State' ($F$), 'Type of Waste' ($W$), '$CO_2$ in Emission' ($C$), 'Filter Efficiency' ($E$), 'Metal in Waste' ($Mi$), 'Light Penetrability' ($L$), 'Emission of Dust' ($D$), and 'Emission of Metal' ($M0$). The variables $B$, $F$, and $W$ are all discrete with states 'stable' or 'unstable', 'intact' or 'defective', and 'industrial' or 'household' respectively. The remaining variables $C$ , $D$ , $E$ , $L$, $Mi$, and $M0$ are all continuous. The junction tree corresponding to the graph of Figure 5.3a is given in Figure 5.3b. This has been constructed by a process of moralisation and weak triangulation and is thus different to the strongly triangulated version given in Lauritzen (1992). The cliques have been marked by ovals and the separators by rectangles.

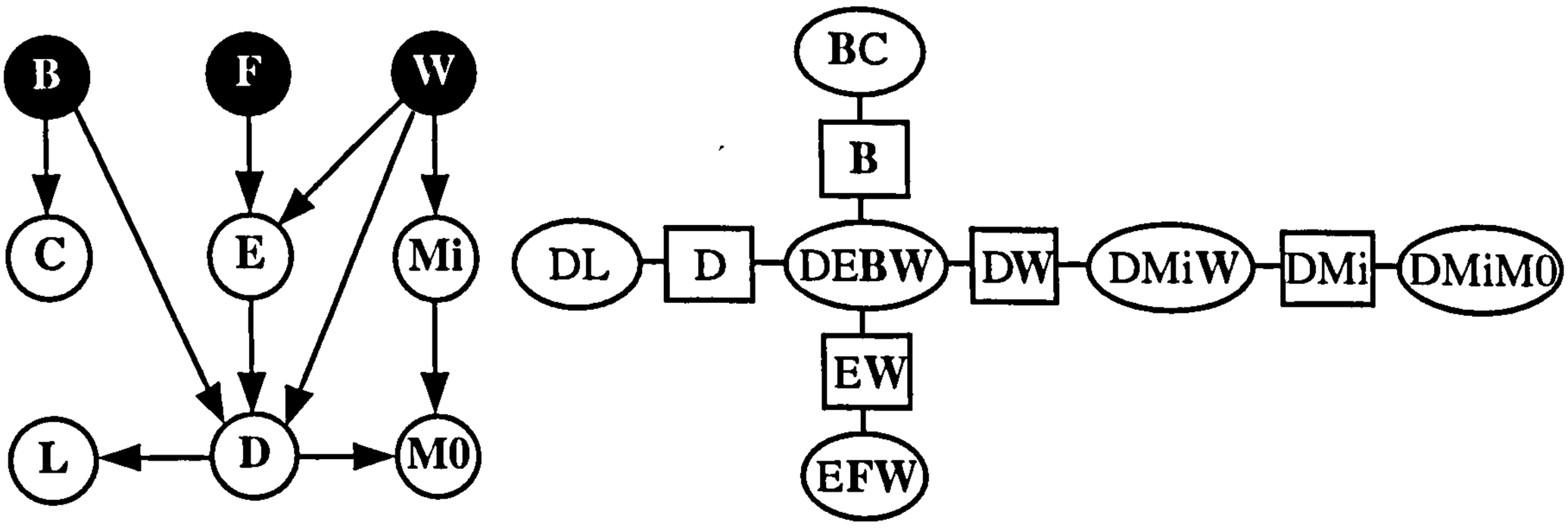



Figure 5.3: Causal probabilistic network (a) and junction tree (b) for the waste incinerator problem.

The marginal probabilities of the discrete variables $B$, $F$ and $W$ and the conditional distributions of the continuous variables $C$, $D$, $E$, $L$, $Mi$ and $M0$ associated with the model are given in Table 4.5.

The continuous variables $C$, $D$, $E$, $L$, $Mi$, $M0$ may be causally ordered: $C$, $E$, $Mi$, $D$, $L$, $M0$. We may thus determine a knot sequence for each of these in turn. Since each of the conditional continuous distributions defined is Normally distributed we shall use a slight modification of the process described earlier to determine the knot sequences for each of the variables. First let us consider where our external knots should be placed. For a Normal distribution $Z_a$ with mean $\mu$ and standard deviation $\sigma$ we may determine that:

$$P(\mu - 3.29053\sigma \leq Z_a \leq \mu + 3.29053\sigma) = 0.999$$

We shall assume that the points $(\mu - 3.29053\sigma)$ and $(\mu + 3.29053\sigma)$ contain enough of the distribution of $Z_a$ for our purposes. For each conditional continuous distribution $f(z_a; i_{pa(a)}, z_{pa(a)})$ we determine these points and choose the minimum and maximum of them, these will be the knots $z_{a,1}$ and $z_{a,M_a-1}$, respectively. Now, using a simple computer programme, we repeat steps **(3)** and **(4)** of our method to determine the number of knots required. For each knot sequence considered the area under the interpolated curve is simply the product of the knot width and the sum of the *internal knots* $z_{a,1}, z_{a,2}, \ldots, z_{a,M_a-1}$. We shall assume that we are satisfied with the univariate spline interpolation of any $f(z_a; i_{pa(a)}, z_{pa(a)})$ if this area is between 0.9999 and 1.0001. This is not a sufficient condition to determine an adequate knot sequence but it is certainly a necessary one. Further testing of the interpolated splines would be optimal but this may, in large networks, prove unwieldy. In general, the more knots used the better the approximation will be but the greater the cost of the approximation. Once all the $f(z_a; i_{pa(a)}, z_{pa(a)})$ have been checked to conform to these rules we place the *external knots*, $z_{a,0}$ and $z_{a,M_a}$, one knot width outside $z_{a,1}$ and $z_{a,M_a-1}$. Since, to satisfy the boundary conditions, the nodal values at the external knots will always be zero, we need not include them in any data structures we may form providing we remember that logically they exist. This can give us a modest saving in space.

Applying the above method we find that two of the conditional distributions, those for $E \mid (F, W)$ and $M0 \mid (D, Mi)$, appear to be troublesome. $M0$ requires in excess of 122 knots while $E$ requires in excess of 494 knots! These results can be seen to arise from the fact that both $E \mid (F, W)$ and $M0 \mid (D, Mi)$ are

| Variable | # Knots | Knot Sequence Range | Knot Width |
|---|---|---|---|
| $C$: $CO_2$ in Emission | 11 | ( -3.5209, 1.2827 ) | 0.480356 |
| $D$: Emission of Dust | 27 | ( 1.1628, 8.4197) | 0.279111 |
| $L$: Light Penetrability | 12 | ( -3.4533, 4.6621 ) | 0.737762 |
| $Mi$: Metal in Waste | 17 | ( -0.8442, 0.9406 ) | 0.111552 |

Table 5.1: The number of knots, ranges of the knots, and knot widths in the knot sequences of each of the continuous variables $C$, $D$, $L$ and $Mi$.

relatively "spiky" distributions. As can be seen from Table 4.5 all the conditional distributions of $E \mid (F, W)$ have exceptionally small variances (between 0.00002 and 0.0001 ). Hence 99.9% of the realisations of $E \mid (F, W)$ will be situated within either 0.0147 or 0.0328 of the corresponding mean. Practically speaking, therefore, these four conditional distributions do not overlap. We thus decide to discretise $E$ into four levels situated around the means $\{-3.9, -3.2, -0.5, -0.4\}$. In the case of $M0$ the variance of $M0 \mid (D, Mi)$ is also quite small (0.002) in relation to the spread of means $(d + mi)$ and 99.9% of each $M0 \mid (D, Mi)$ is within 0.1472 of the corresponding mean. Using the knot sequences defined for $D$ and $Mi$ we are able to put some sensible bounds on the means of the conditional distributions $M0 \mid (D,\ Mi)$ - these are $(0.23803, 9.3443)$. We shall choose to discretise $M0$ into 40 intervals for the purposes of this investigation. The conditional distribution $M0 \mid (D = d,\ Mi = mi)$ is set equal to one in the interval containing $(d + mi)$ and zero elsewhere.

The numbers of knots, knot sequence ranges, and knot widths of the four continuous distributions $C$, $D$, $L$ and $Mi$ are given in Table 5.1. It should be remembered that only the 9, 25, 10 and 15 internal knots of $C$, $D$, $L$ and $Mi$, respectively, are actually needed to perform the calculations.

Having determined the knot sequences we may determine the nodal values of the conditional distributions. Consider, for example, the conditional distribution for $Z_a \mid X_{pa(a)}$ defined as follows:

$$Z_a \mid X_{pa(a)} = Z \mid (I_{pa(a)}, Z_{pa(a)}) \sim N(\alpha(i_{pa(a)}) + \beta(i_{pa(a)})^T z_{pa(a)}, \sigma^2(i_{pa(a)}))$$

where:

$$\beta(i_{pa(a)})^T = (\beta_1(i_{pa(a)}), \beta_2(i_{pa(a)}), \ldots, \beta_n(i_{pa(a)}))$$

and each $\beta_j(i_{pa(a)})$ corresponds to a continuous variable $Z_j$ such that $j \in pa(a)$. Then we may determine the nodal values of the conditional distribution of $Z_a \mid X_{pa(a)}$ at a node $(z_a, z_{pa(a)})$ to be:

$$\begin{aligned}
f(i_a, z_a, z_{pa(a)}) &= \frac{1}{\sqrt{2\pi\sigma^2(i_a)}} \exp\left\{-\frac{1}{2\sigma^2(i_a)}\left(z_a - \alpha(i_a) - \beta(i_a)^T z_{pa(a)}\right)^2\right\} \\
\frac{\delta f(i_a, z_a, z_{pa(a)})}{\delta z_a} &= \frac{1}{\sigma^2(i_a)}\left(\alpha(i_a) + \beta(i_a)^T z_{pa(a)} - z_a\right) f(i_a, z_a, z_{pa(a)}) \\
\frac{\delta f(i_a, z_a, z_{pa(a)})}{\delta z_j} &= \frac{\beta_j(i_a)}{\sigma^2(i_a)}\left(z_a - \alpha(i_a) - \beta(i_a)^T z_{pa(a)}\right) f(i_a, z_a, z_{pa(a)}) \\
&\quad \text{for } j \in pa(a).
\end{aligned}$$

Each set of nodal values should then, preferably, be normalised. This will ensure that each continuous distribution marginalises to one and not to some value between 0.9999 and 1.0001. This normalisation will thus prevent our spline interpolated continuous distributions from polluting the discrete variables in our network and will ensure that any normalisation constant determined after the addition of evidence is simply the probability of obtaining that evidence.

### 5.7.5 Initialisation

Associated with each clique $C \in \mathcal{C}$ is a potential function $a_C(x_C)$ which is a non-negative function defined on the variables, $x_C$, in that clique. Similarly associated with each separator $S \in \mathcal{S}$ we have a potential function $b_S(x_S)$, which is a non-negative function defined on the variables, $x_S$, in that separator. We initially let $a_C(x_C) \equiv 1$, for all $C \in \mathcal{C}$ and $b_S(x_S) \equiv 1$, for all $S \in \mathcal{S}$. Associated with each potential function $a_C(x_C)$ (or $b_S(x_S)$) there is a potential table. Each cell in the potential table corresponds to a particular combination of the levels of the discrete and continuous variables. The contents of each cell is a set of nodal values for that particular combination of variables. Initially these nodal values are $(1, 0, \ldots, 0)$ where the one corresponds to the function value at the node and the zeros correspond to the first derivatives at that node with respect to each of the continuous variables. When combined with the knots, the potential tables contain all the information necessary to form the potential functions.

For each variable $X_a$, for $a \in K$, we assign $X_a$ to a clique $C$ in the junction tree. We may only assign $X_a$ to cliques which contain all the variables necessary to define the conditional distribution of $X_a$ given its parents $X_{pa(a)}$. If more than one suitable clique exists then the choice of which one of them $X_a$ is to be assigned to is arbitrary. When a variable $X_a$ is assigned to a clique $C$ the potential table, $a_C(x_C)$, on that clique is updated by multiplying it by our representation for the conditional probability density function of $X_a \mid X_{pa(a)}$ (the multiplication of potential functions will be defined later). When all the variables have been assigned $Pf$, our interpolation of the true joint probability density function $f$ of $X$, is as follows:

$$Pf_K(x) = \prod_{C \in \mathcal{C}} a_C(x_C) \tag{5.35}$$

since $f$ factorises into the conditional distributions of all the variables given their parents. We may trivially express Equation 5.35 as:

$$Pf_K(x) = \frac{\prod_{C \in \mathcal{C}} a_C(x_C)}{\prod_{S \in \mathcal{S}} b_S(x_S)} \tag{5.36}$$

since $b_S \equiv 1$ (we shall define the division of potential functions later). Equation 5.36 is termed the *joint system belief*.

The assignment of variables to cliques in the Waste Incinerator Problem is given in Table 4.6.

### 5.7.6 Propagation

The joint system belief we have constructed in Equation 5.36 is the correct interpolated joint probability density function of all the variables in our model. However, we do not, necessarily, have the correct interpolated joint probability density function of all the variables in any given clique (separator) represented by the potential function associated with that clique (separator). In order to achieve this we must make the potentials on the cliques and separators consistent by passing a propagation schedule composed of a series of active flows. We use Definition 23 to define the passage of a flow. The basic operations required to pass a flow will be discussed in the next section. The propagation schedule for the waste incinerator problem is given in Table 4.8.

## 5.8 Potential Tables

Suppose we have a finite set of vertices $K$ of a graph $\mathcal{G} = (K, E)$ partitioned into discrete, $\Delta$, and continuous, $\Omega$, types as $K = \Delta \cup \Omega$. Let $X = X_K$ denote the variables corresponding to these vertices and suppose that $I = I_\Delta$ and $Z = Z_\Omega$ represent the discrete and continuous variables in $X$ respectively. Suppose that each continuous variable $Z_a$, for $a \in \Omega$, has been discretised and interpolated using the techniques described earlier. Now, let us consider some subset of these vertices $V \subseteq K$ with related variables $X_V = (I_V, Z_V)$ where $I_V$ and $Z_V$ are the discrete and continuous variables in $X_V$ respectively. Without loss of generality we will label the variables $Z_a$, for $a \in V \cap \Omega$, as $Z_V = (Z_1, Z_2, \ldots, Z_{q_V})$. We shall assume that we have some underlying potential functions $\phi$ and $\eta$ on the variables $X_V$. We may then characterise $\phi$ and $\eta$ using potential tables consisting of the nodal values of $\phi$ and $\eta$. We shall let $\phi^*$ and $\eta^*$ denote the potential tables for $\phi$ and $\eta$ respectively, and we may thus write:

$$\begin{aligned}
\phi^*(x_V) &= \phi^*(i_V, z_V) = \left\{\phi(i_V, z_V), \frac{\delta\phi(i_V, z_V)}{\delta z_1}, \frac{\delta\phi(i_V, z_V)}{\delta z_2}, \ldots, \frac{\delta\phi(i_V, z_V)}{\delta z_{q_V}}\right\} \\
\eta^*(x_V) &= \eta^*(i_V, z_V) = \left\{\eta(i_V, z_V), \frac{\delta\eta(i_V, z_V)}{\delta z_1}, \frac{\delta\eta(i_V, z_V)}{\delta z_2}, \ldots, \frac{\delta\eta(i_V, z_V)}{\delta z_{q_V}}\right\}
\end{aligned} \tag{5.37}$$

We shall now consider how potential tables are affected by the basic operations required to pass an absorption. These basic operations are *marginalisation*, *extension*, *multiplication* and *division*.

### 5.8.1 Marginalisation over Discrete Variables

**Theorem 37** *Suppose that $X_V$ is a set of random variables such that $X_V = (I_V, Z_V)$, where $I_V$ is a set of discrete variables, and $Z_V$ is a set of continuous variables. Let $\phi^*(x_V)$, the potential table for $X_V$, be defined as in Equation 5.37. Let $U \subseteq V$ and suppose that we may partition $I_V$ into two disjoint sets $I_U$ and $I_{V \setminus U}$. Also, let $Z_U = Z_V$. Suppose that we wish to marginalise $\phi^*(x_V)$ with respect to $I_{V \setminus U}$. The potential table for $(I_U, Z_V)$ formed by this marginalisation is denoted $\eta^*_U$ and the nodal values associated with a particular cell of this potential table when the levels of $(I_U, Z_V)$ are $(i_U, z_V)$ are as follows:*

$$\eta^*{}_U(i_U, z_V) = \left\{ \sum_{I_{V\setminus U}} \phi(i_V, z_V), \sum_{I_{V\setminus U}} \frac{\delta\phi(i_V, z_V)}{\delta z_1}, \ldots, \sum_{I_{V\setminus U}} \frac{\delta\phi(i_V, z_V)}{\delta z_{q_V}} \right\} \tag{5.38}$$

**Proof.** Using our definition of $\phi^*(x_V)$ as given in Equations 5.37 and noting that the operation of differentiation is distributive over summation we find that:

$$\begin{aligned}
\eta^*{}_U(i_U, z_V) &= \sum_{I_{V\setminus U}} \phi^*(i_V, z_V) \\
&= \left\{ \sum_{I_{V\setminus U}} \phi(i_V, z_V), \frac{\delta \sum_{I_{V\setminus U}} \phi(i_V, z_V)}{\delta z_1}, \ldots, \frac{\delta \sum_{I_{V\setminus U}} \phi(i_V, z_V)}{\delta z_{q_V}} \right\} \\
&= \left\{ \sum_{I_{V\setminus U}} \phi(i_V, z_V), \sum_{I_{V\setminus U}} \frac{\delta\phi(i_V, z_V)}{\delta z_1}, \ldots, \sum_{I_{V\setminus U}} \frac{\delta\phi(i_V, z_V)}{\delta z_{q_V}} \right\}
\end{aligned}$$

□

### 5.8.2 Marginalisation over Continuous Variables

**Theorem 38** *Suppose that $X_V$ is a set of random variables such that $X_V = (I_V, Z_V)$, where $I_V$ is a set of discrete variables, and $Z_V$ is a set of continuous variables. Let $\phi^*(x_V)$, the potential table for $X_V$, be defined as in Equation 5.37. Let $U \subseteq V$ and suppose that we may partition $Z_V$ into two disjoint sets $Z_U$ and $Z_{V\setminus U}$. Also, let $I_U = I_V$. Without loss of generality let us label the $q_U$ variables in $Z_U$ as $(Z_1, Z_2, \ldots, Z_{q_U})$ and the $q_{V\setminus U}$ variables in $Z_{V\setminus U}$ as $(Z_{q_U+1}, Z_{q_U+2}, \ldots, Z_{q_V})$. Suppose that we wish to marginalise $\phi^*(x_V)$ with respect to $Z_{V\setminus U}$. The potential table for $(I_V, Z_U)$ formed by this marginalisation is denoted $\eta^*{}_U$ and the nodal values associated with a particular cell of this potential table when the levels of $(I_V, Z_U)$ are $(i_V, z_U)$ are as follows:*

$$\eta^*{}_U(i_V, z_U) = \left\{ h \sum_{V\setminus U} \phi(i_V, z_V), h \sum_{V\setminus U} \frac{\delta\phi(i_V, z_V)}{\delta z_1}, \ldots, h \sum_{V\setminus U} \frac{\delta\phi(i_V, z_V)}{\delta z_{q_U}} \right\}$$

*where:*

$$h = \prod_{a\in V\setminus U} h_a \tag{5.39}$$

*and $h_a$ is the knot width of the knot sequence defined for a variable $Z_a$, for $a \in V\setminus U$.*

**Proof.** We showed in Theorem 36 how the cubic spline interpolant for a set of continuous variables may be formed when marginalising over a continuous variable $X_j$, say, by dropping any derivative data on $X_j$, summing the remaining variables across the levels of $X_j$ and multiplying them by the knot width of $X_j$. When marginalising over a set of continuous variables $Z_{V\setminus U}$, say, we simply repeat these operations over each continuous variable in any order in turn and derive the result given in Equation 5.39.

□

### 5.8.3 Extension

Let $U \subseteq V \subseteq K$ and, without loss of generality, let us label the $q_U$ variables in $Z_U$ as $(Z_1, Z_2, \ldots, Z_{q_U})$ and the $q_{V\setminus U}$ variables in $Z_{V\setminus U}$ as $(Z_{q_U+1}, Z_{q_U+2}, \ldots, Z_{q_V})$. Let $\phi^*(x_U) = \phi^*(i_U, z_U)$ be the potential table for a potential function $\phi(i_U, z_U)$ defined on $\mathcal{X}_U = \mathcal{I}_U \times \mathcal{Z}_U$, where:

$$\phi^*(x_U) = \phi^*(i_U, z_U) = \left\{\phi(i_U, z_U), \frac{\delta\phi(i_U, z_U)}{\delta z_1}, \ldots, \frac{\delta\phi(i_U, z_U)}{\delta z_{q_U}}\right\}$$

Now suppose that $\phi^*(x_U)$ is extended to $\eta^*(x_V)$ defined on $\mathcal{X}_V = \mathcal{I}_V \times \mathcal{Z}_V = (\mathcal{I}_U \times \mathcal{I}_{V\setminus U}) \times (\mathcal{Z}_U \times \mathcal{Z}_{V\setminus U})$ then if $\mathcal{Z}_{V\setminus U} \in \mathbb{R}^{q_{V\setminus U}}$ the new nodal values defined on $x_V$ will be:

$$\eta^*(x_V) = \eta^*\left(i_U, i_{V\setminus U}, z_U, z_{V\setminus U}\right) = \left\{\phi(i_U, z_U), \frac{\delta\phi(i_U, z_U)}{\delta z_1}, \ldots, \frac{\delta\phi(i_U, z_U)}{\delta z_{q_U}}, (0)\right\}$$

where (0) is a row vector of length $q_{V\setminus U}$ corresponding to the $q_{V\setminus U}$ first derivatives of $\phi(i_U, z_U)$ with respect to the variables in $Z_{V\setminus U}$.

### 5.8.4 Multiplication

If $\phi^*$ and $\eta^*$ are the potential tables of two potential functions which have been extended to occupy the same space, $\mathcal{X}_V$, and $\phi^*$ and $\eta^*$ are defined as in Equations 5.37, then multiplication is defined, by use of the product rule, as:

$$(\phi^* \times \eta^*)(x_V) =$$

$$\left\{\phi(i_V, z_V) \times \eta(i_V, z_V), \frac{\delta(\phi(i_V, z_V) \times \eta(i_V, z_V))}{\delta z_1}, \ldots, \frac{\delta(\phi(i_V, z_V) \times \eta(i_V, z_V))}{\delta z_{q_V}}\right\}$$

where, for $a = 1, \ldots, q_V$:

$$\frac{\delta\left(\phi(i_V, z_V) \times \eta(i_V, z_V)\right)}{\delta z_a} = \left(\phi(i_V, z_V) \times \frac{\delta\eta(i_V, z_V)}{\delta z_a}\right) + \left(\frac{\delta\phi(i_V, z_V)}{\delta z_a} \times \eta(i_V, z_V)\right)$$

It should be noted that, in general, the multiplication of two potential functions $\phi^*$ and $\eta^*$ will not result in the formation of a third potential function of the same type (i.e. a Hermitian cubic spline may not be formed). In a PES, however, we are fortunate in the fact that initially the projector for each variable is contained only once in the junction tree. Therefore whenever potential functions are multiplied together we are essentially concerned with the multiplication of a new projector orthogonal to all the others which, as seen in Section 5.4, will always result in potential functions of the same type. Thus the form of our potential functions will be retained by multiplication. A similar argument applies to the division of two potential functions.

### 5.8.5 Division

If $\phi^*$ and $\eta^*$ are the potential tables of two potential functions which have been extended to occupy the same space, $\mathcal{X}_V$, and $\phi^*$ and $\eta^*$ are defined as in Equations 5.37, then division is defined, by use of the quotient rule, as follows:

$$(\phi^*/\eta^*)(x_V) = \begin{cases} \{0, 0, \ldots, 0\} & : \quad \phi(i_V, z_V) = 0 \\ \left\{\frac{\phi(i_V, z_V)}{\eta(i_V, z_V)}, \frac{\delta\left(\frac{\phi(i_V, z_V)}{\eta(i_V, z_V)}\right)}{\delta z_1}, \ldots, \frac{\delta\left(\frac{\phi(i_V, z_V)}{\eta(i_V, z_V)}\right)}{\delta z_{q_V}}\right\} & : \quad \eta(i_V, z_V) \neq 0 \\ \text{undefined} & : \quad \text{otherwise} \end{cases}$$

where, for $a = 1, \ldots, q_V$:

$$\frac{\delta\left(\frac{\phi(i_V, z_V)}{\eta(i_V, z_V)}\right)}{\delta z_a} = \frac{\left(\frac{\delta\phi(i_V, z_V)}{\delta z_a} \times \eta(i_V, z_V)\right) - \left(\phi(i_V, z_V) \times \frac{\delta\eta(i_V, z_V)}{\delta z_a}\right)}{\left(\eta(i_V, z_V)\right)^2}$$

## 5.9 Determining Moments

After propagation we may determine the marginal distributions of each continuous variable. We shall now consider how we may determine the moment characteristics of these distributions. Our information on a marginal distribution of a particular continuous random variable $Z_a$, for $a \in \Omega$, consists of a knot sequence $(z_{a,0}, z_{a,1}, \ldots, z_{a,M_a})$ with data $(g_{a,m_a}, g'_{a,m_a})$, for $m_a = 0, 1, \ldots, M_a$, representing the 0th and 1st derivatives of the distribution at each of these knots. Our cubic spline interpolation $p_{a,m_a}(z_a)$ for the distribution in the interval $[z_{a,m_a}, z_{a,m_a+1}]$ is as given in Equation 5.6 for $Z = Z_a$. We may determine the mean of $Z_a$ to be as follows:

$$
\begin{aligned}
E[Z_a] &= \int_{z_a=-\infty}^{z_a=+\infty} z_a g_a(z_a) dz_a \quad \cong \quad \sum_{m_a=0}^{M_a-1} \int_{z_{a,m_a}}^{z_{a,m_a+1}} z_a p_{m_a}(z_a) dz_a \\
&= \sum_{m_a=0}^{M_a-1} \int_{z_{a,m_a}}^{z_{a,m_a+1}} z_a \Bigg\{ \left( \frac{2(g_{a,m_a} - g_{a,m_a+1})}{h_{a,m_a}^3} + \frac{(g'_{a,m_a} + g'_{a,m_a+1})}{h_{a,m_a}^2} \right) (z_a - z_{a,m_a})^3 \\
&\qquad + \left( \frac{3(g_{a,m_a+1} - g_{a,m_a})}{h_{a,m_a}^2} - \frac{(2g'_{a,m_a} + g'_{a,m_a+1})}{h_{a,m_a}} \right) (z_a - z_{a,m_a})^2 \\
&\qquad + g'_{a,m_a}(z_a - z_{a,m_a}) + g_{a,m_a} \Bigg\} dz_a \\
&= \frac{h_a}{60} \sum_{m_a=0}^{M_a-1} \Big\{ \left( 2g'_{a,m_a+1} - 3g'_{a,m_a} \right) z_{a,m_a}^2 + \left( 21g_{a,m_a} + 9g_{a,m_a+1} \right) z_{a,m_a} \\
&\qquad + \left( g'_{a,m_a} + g'_{a,m_a+1} \right) z_{a,m_a} z_{a,m_a+1} \\
&\qquad + \left( 9g_{a,m_a} + 21g_{a,m_a+1} \right) z_{a,m_a+1} + \left( 2g'_{a,m_a} - 3g'_{a,m_a+1} \right) z_{a,m_a+1}^2 \Big\}
\end{aligned}
\tag{5.40}
$$

Similarly $E[Z_a^2]$ may be determined thus:

$$
\begin{aligned}
E[Z_a^2] &= \int_{z_a=-\infty}^{z_a=+\infty} z_a^2 g_a(z_a) dz_a \quad \cong \quad \sum_{m_a=0}^{M_a-1} \int_{z_{a,m_a}}^{z_{a,m_a+1}} z_a^2 p_{m_a}(z_a) dz_a \\
&= \sum_{m_a=0}^{M_a-1} \int_{z_{a,m_a}}^{z_{a,m_a+1}} z_a^2 \Bigg\{ \left( \frac{2(g_{a,m_a} - g_{a,m_a+1})}{h_{a,m_a}^3} + \frac{(g'_{a,m_a} + g'_{a,m_a+1})}{h_{a,m_a}^2} \right) (z_a - z_{a,m_a})^3 \\
&\qquad + \left( \frac{3(g_{a,m_a+1} - g_{a,m_a})}{h_{a,m_a}^2} - \frac{(2g'_{a,m_a} + g'_{a,m_a+1})}{h_{a,m_a}} \right) (z_a - z_{a,m_a})^2 \\
&\qquad + g'_{a,m_a}(z_a - z_{a,m_a}) + g_{a,m_a} \Bigg\} dz_a
\end{aligned}
$$

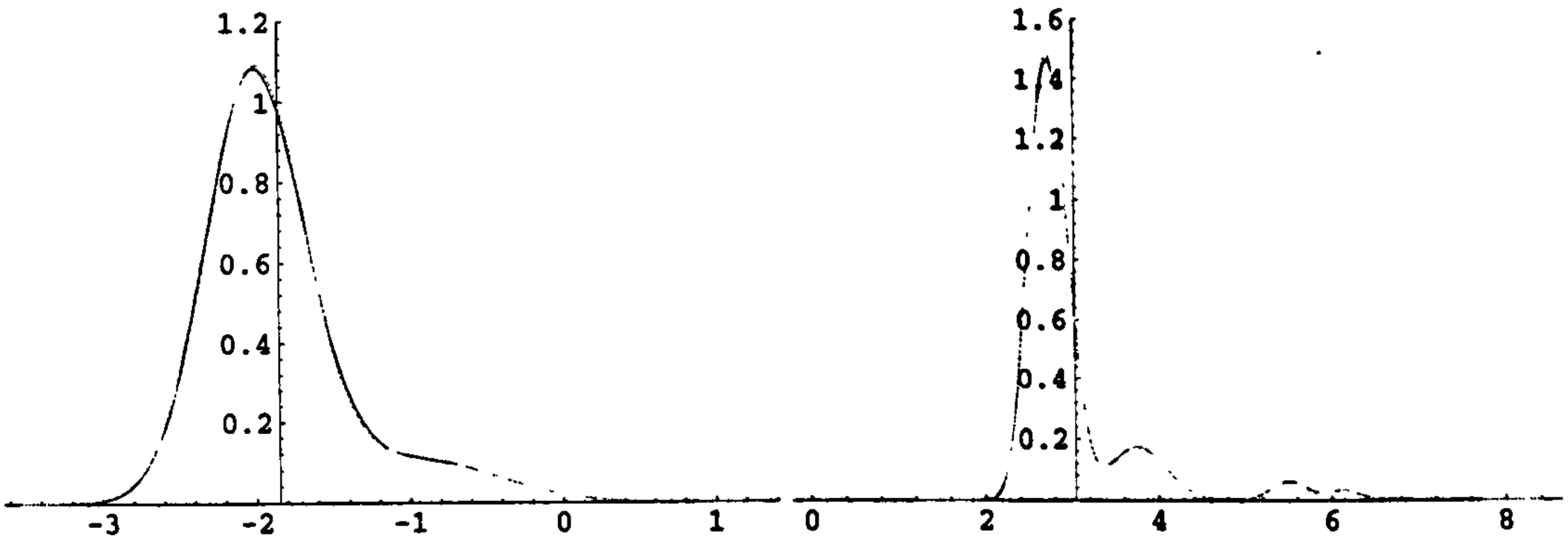
$$= \frac{h_a}{60}\sum_{m_a=0}^{M_a-1}\Big\{\left(g'_{a,m_a+1}-2g'_{a,m_a}\right)z^3_{a,m_a}+4\left(4g_{a,m_a}+g_{a,m_a+1}\right)z^2_{a,m_a}$$
$$+\left(g'_{a,m_a+1}\right)z^2_{a,m_a}z_{a,m_a+1}+10\left(g_{a,m_a}+g_{a,m_a+1}\right)z_{a,m_a}z_{a,m_a+1}$$
$$+\left(g'_{a,m_a}\right)z_{a,m_a}z^2_{a,m_a+1}+4\left(g_{a,m_a}+4g_{a,m_a+1}\right)z^2_{a,m_a+1}$$
$$+\left(g'_{a,m_a}-2g'_{a,m_a+1}\right)z^3_{a,m_a+1}\Big\} \tag{5.41}$$

Combining Equations 5.40 and 5.41 we are able to determine the variance of $Z_a$ using the well known result $\mathrm{Var}(Z_a) = E[Z_a^2] - (E[Z_a])^2$.

| Variable | Spline Interpolated | Exact |
|---|---|---|
| $B$ | (0.850000, 0.150000) | (0.850000, 0.150000) |
| $E$ | (0.271429, 0.678571, 0.0357143, 0.0142857) | (0.271429, 0.678571, 0.0357143, 0.0142857) |
| $F$ | (0.950000, 0.0500000) | (0.950000, 0.0500000) |
| $W$ | (0.285714, 0.714286) | (0.285714, 0.714286) |
| $C$ | (-1.85032, 0.257314) | (-1.85000, 0.257500) |
| $D$ | (3.03921, 0.592947) | (3.03929, 0.592885) |
| $L$ | (1.48052, 0.397929) | (1.48036, 0.398221) |
| $Mi$ | (-0.214350, 0.210516) | (-0.214286, 0.210511) |
| $M0$ | (2.82905, 0.756075) | (2.82500, 0.740089) |

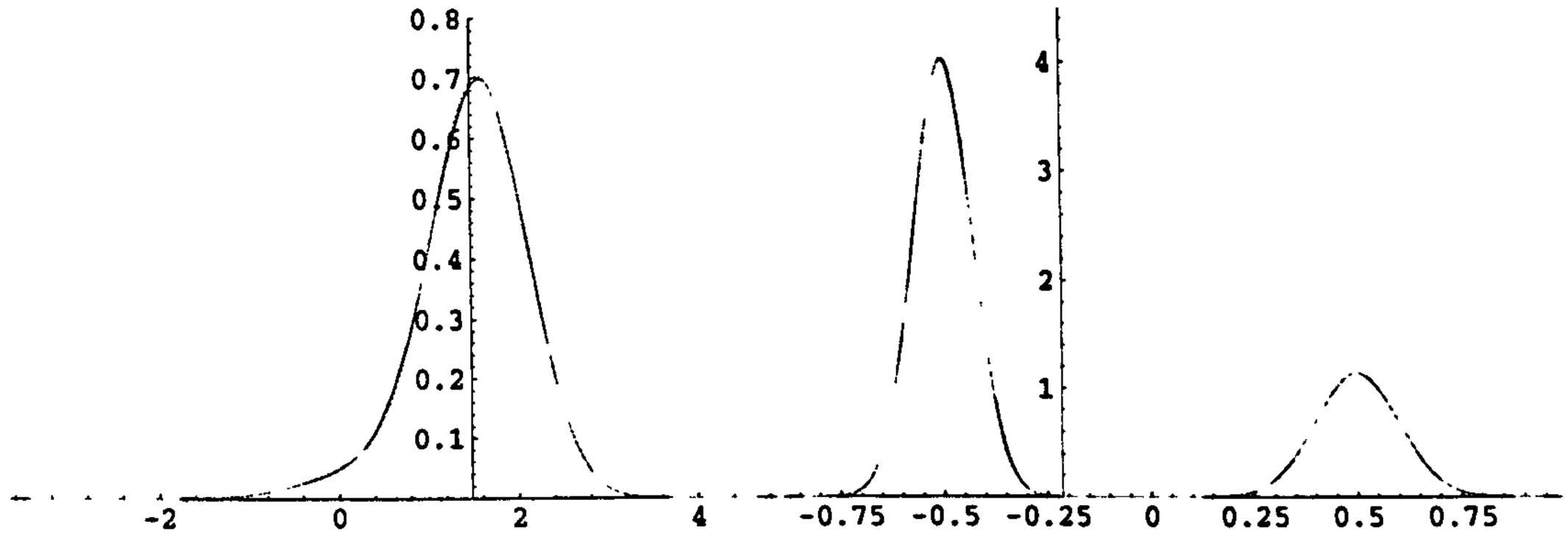
Table 5.2: Means, variances and probabilities of the marginal distributions for the waste incinerator problem.

The means and variances of the marginal distributions of the continuous random variables, and the marginal probabilities of the discrete random variables in the waste incinerator problem are given in Table 5.2 for the exact and spline interpolated cases. The marginal distributions of the continuous random variables in the waste incinerator example given no evidence are graphed in Figure 5.4 for the exact and spline interpolated cases. The graph of $M0$ has been formed by putting the marginal probability divided by the interval width at the midpoints of each interval and fitting a spline curve to the result.



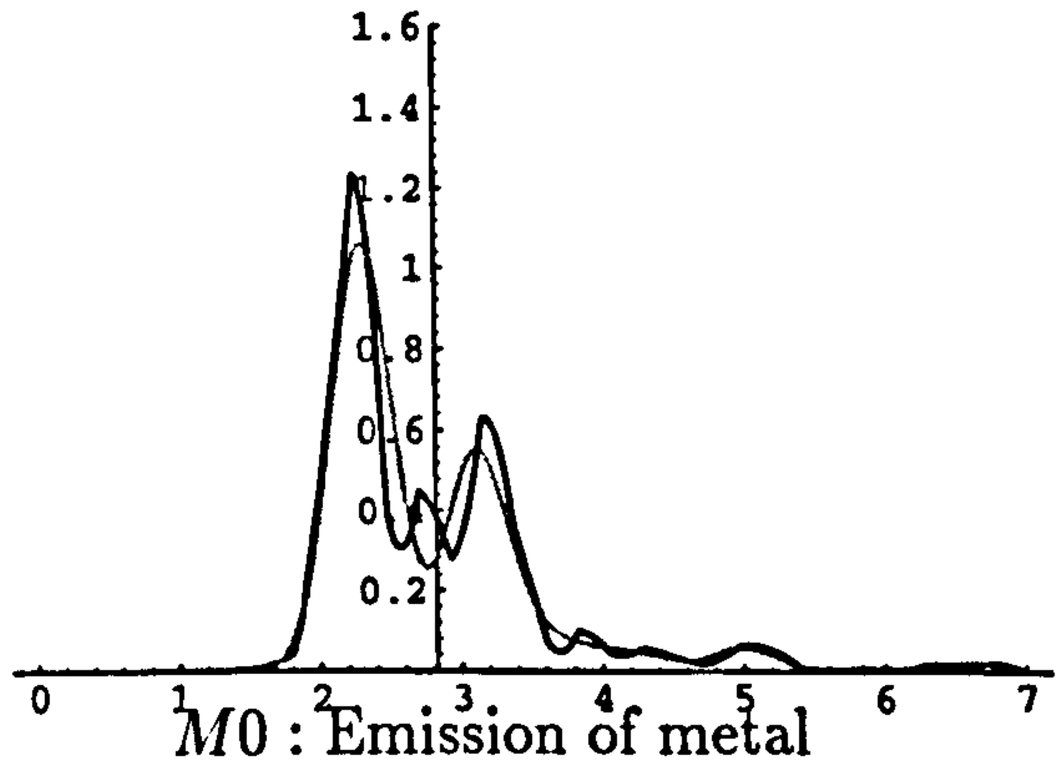

$C$ : $CO_2$ in emission

$D$ : Dust in emission




$L$ : Light penetrability

$Mi$ : Metal in waste




$M0$ : Emission of metal

Figure 5.4: Graphs of the exact and spline interpolated marginal distributions of the continuous variables in the waste incinerator problem. Exact distributions are given in grey, spline interpolated ones in black.

## 5.10 Adding Evidence

We shall now consider how we may add evidence $\mathcal{E}$ to our system. We shall assume that evidence on a discrete variable informs us that a particular variable is at a given state and we wish to update our uncertainty appropriately. We do this by reducing the potential tables in each clique and each separator to consist of cells at this known state only. For example, suppose that we have the potential table $\phi^*$ given in Equation 5.37 and without loss of generality let $I_V = (I_1, I_2, \ldots, I_{p_V})$ and $Z_V = (Z_1, Z_2, \ldots, Z_{q_V})$ then the updated potential table $\phi^*_{\mathcal{E}}$ once the evidence $\mathcal{E} : I_\delta = e$, for $\delta \in \{1, 2, \ldots, p_V\}$, has been entered is as follows:

$$\begin{aligned}
\phi^*_{\mathcal{E}}\left(x_{V|I_\delta=e}\right) &= \phi^*_{\mathcal{E}}\left(i_1, \ldots, i_{\delta-1}, i_{\delta+1}, \ldots, i_{p_V}, z_V\right) \\
&= \left\{\phi\left(i_1, \ldots, i_{\delta-1}, e, i_{\delta+1}, \ldots, i_{p_V}, z_V\right),\right. \\
&\quad \frac{\delta\phi\left(i_1, \ldots, i_{\delta-1}, e, i_{\delta+1}, \ldots, i_{p_V}, z_V\right)}{\delta z_1}, \ldots, \\
&\quad \left.\frac{\delta\phi\left(i_1, \ldots, i_{\delta-1}, e, i_{\delta+1}, \ldots, i_{p_V}, z_V\right)}{\delta z_{q_V}}\right\}
\end{aligned} \tag{5.42}$$

We shall assume that evidence on a continuous variable informs us that that variable is a particular value. We must enter this information into every clique and separator which contains the known variable. In the case where the evidence on a continuous variable lies at a knot defined for that variable we may simply select those cells of the potential table which correspond to that knot and include them, and them alone, in the updated potential table. At the same time we must drop any derivative data on that variable. Suppose we have the potential table $\phi^*$ given in Equation 5.37 and without loss of generality let $Z_V = (Z_1, Z_2, \ldots, Z_{q_V})$ and let $e$ be a knot for variable $Z_a$. Then the updated potential table $\phi^*_{\mathcal{E}}$ once the evidence $\mathcal{E} : Z_a = e$, for $a \in \{1, 2, \ldots, q_V\}$, has been entered is as follows:

$$\begin{aligned}
\phi^*_{\mathcal{E}}\left(x_{V|Z_a=e}\right) &= \phi^*_{\mathcal{E}}\left(i_V, z_1, \ldots, z_{a-1}, z_{a+1}, \ldots, z_{q_V}\right) \\
&= \left\{\phi\left(i_V, z_1, \ldots, z_{a-1}, e, z_{a+1}, \ldots, z_{q_V}\right),\right. \\
&\quad \left.\frac{\delta\phi\left(i_V, z_1, \ldots, z_{a-1}, e, z_{a+1}, \ldots, z_{q_V}\right)}{\delta z_1}, \ldots,\right.
\end{aligned}$$

$$\left.\begin{array}{c}\dfrac{\delta\phi(i_V, z_1, \ldots, z_{a-1}, e, z_{a+1}, \ldots, z_{q_V})}{\delta z_{a-1}}, \\ \dfrac{\delta\phi(i_V, z_1, \ldots, z_{a-1}, e, z_{a+1}, \ldots, z_{q_V})}{\delta z_{a+1}}, \ldots, \\ \dfrac{\delta\phi(i_V, z_1, \ldots, z_{a-1}, e, z_{a+1}, \ldots, z_{q_V})}{\delta z_{q_V}}\end{array}\right\} \tag{5.43}$$

In the case where the evidence $\mathcal{E} : Z_a = e$ on a particular continuous random variable $Z_a$ is not situated at a knot then we must interpolate using the information on the knots bounding the interval that contains that evidence. Suppose we have the potential table $\phi^*$ given in Equation 5.37 and without loss of generality let $Z_V = (Z_1, Z_2, \ldots, Z_{q_V})$ and let $z_{a,m_a} \leq e \leq z_{a,m_a+1}$ where $z_{a,m_a}$ and $z_{a,m_a+1}$ are neighbouring knots in the interpolation of $Z_a$. Then the updated potential table $\phi^*_{\mathcal{E}}$ once the evidence that $Z_a = e$, for $a \in \{1, 2, \ldots, q_V\}$, has been entered may be written as in Equation 5.43 where:

$$\begin{aligned}\phi(i_V, z_1, \ldots, z_{a-1}, e, z_{a+1}, \ldots, z_{q_V}) \;=\; & \\ & \left(\frac{2(\phi^*(z_{a,m_a}) + \phi^*(z_{a,m_a+1}))}{h_a^3} + \frac{(\phi'_a(z_{a,m_a}) + \phi'_a(z_{a,m_a+1}))}{h_a^2}\right)(e - z_{a,m_a})^3 \\ & + \left(\frac{3(\phi^*(z_{a,m_a+1}) - \phi^*(z_{a,m_a}))}{h_a^2} - \frac{(2\phi'_a(z_{a,m_a}) + \phi'_a(z_{a,m_a+1}))}{h_a}\right)(e - z_{a,m_a})^2 \\ & + \phi'_a(z_{a,m_a})(e - z_{a,m_a}) + \phi^*(z_{a,m_a})\end{aligned}$$

$$\begin{aligned}\frac{\delta\phi(i_V, z_1, \ldots, z_{a-1}, e, z_{a+1}, \ldots, z_{q_V})}{\delta z_j} \;=\; & \\ & \phi'_j(z_{a,m_a}) + \left(\frac{\phi'_j(z_{a,m_a+1}) - \phi'_j(z_{a,m_a})}{h_a}\right)(e - z_{a,m_a})\end{aligned}$$

$$\text{for } j = 1, \ldots, a-1, a+1, \ldots, q_V$$

and:

$$\begin{aligned} h_a &= (z_{a,m_a+1} - z_{a,m_a}) \\ \phi^*(z_{a,m_a}) &= \phi(i_V, z_1, \ldots, z_{a-1}, z_{a,m_a}, z_{a+1}, \ldots, z_{q_V}) \\ \phi^*(z_{a,m_a+1}) &= \phi(i_V, z_1, \ldots, z_{a-1}, z_{a,m_a+1}, z_{a+1}, \ldots, z_{q_V}) \end{aligned}$$

$$
\begin{aligned}
\phi'_a(z_{a,m_a}) &= \frac{\delta\phi(i_V, z_1, \ldots, z_{a-1}, z_{a,m_a}, z_{a+1}, \ldots, z_{q_V})}{\delta z_a} \\
\phi'_a(z_{a,m_a+1}) &= \frac{\delta\phi(i_V, z_1, \ldots, z_{a-1}, z_{a,m_a+1}, z_{a+1}, \ldots, z_{q_V})}{\delta z_a} \\
\phi'_j(z_{a,m_a}) &= \frac{\delta\phi(i_V, z_1, \ldots, z_{a-1}, z_{a,m_a}, z_{a+1}, \ldots, z_{q_V})}{\delta z_j} \\
\phi'_j(z_{a,m_a+1}) &= \frac{\delta\phi(i_V, z_1, \ldots, z_{a-1}, z_{a,m_a+1}, z_{a+1}, \ldots, z_{q_V})}{\delta z_j}
\end{aligned} \tag{5.44}
$$

Here the function values have been interpolated using cubic spline interpolation while the derivatives are interpolated linearly since derivative data higher than the first derivatives is not available.

Having entered a collection of evidence $\mathcal{E}$ into the system we must pass a propagation schedule. This results in the joint system belief being equal to the interpolated joint probability density function of the variables given the evidence $\mathcal{E}$. Normalising the cliques and separators we are able to obtain the updated joint system belief and hence we may determine the marginals.

Lauritzen (1992) considers the addition of the evidence $\mathcal{E} : \{W = \text{Industrial}, C = -0.9, L = 1.1\}$ into the waste incinerator example while Olesen (1991) considers the addition of the evidence $\mathcal{E} : \{W = \text{Industrial}, C = -1.6\ , L = 0.5\}$. The probabilities, means and variances of the marginal distributions of the unknown variables given Lauritzen's evidence are presented in Table 5.3, and they are graphed in Figure 5.5. The probabilities, means and variances of the marginal distributions of the unknown variables given Olesen's evidence are presented in Table 5.4, and they are graphed in Figure 5.6. Both the exact and spline interpolated cases are presented. The exact case has been calculated with the discretisation of the continuous random variable $E$ to add comparison. Graphs relating to the exact case are given in grey and those relating to the spline interpolated case are given in black.

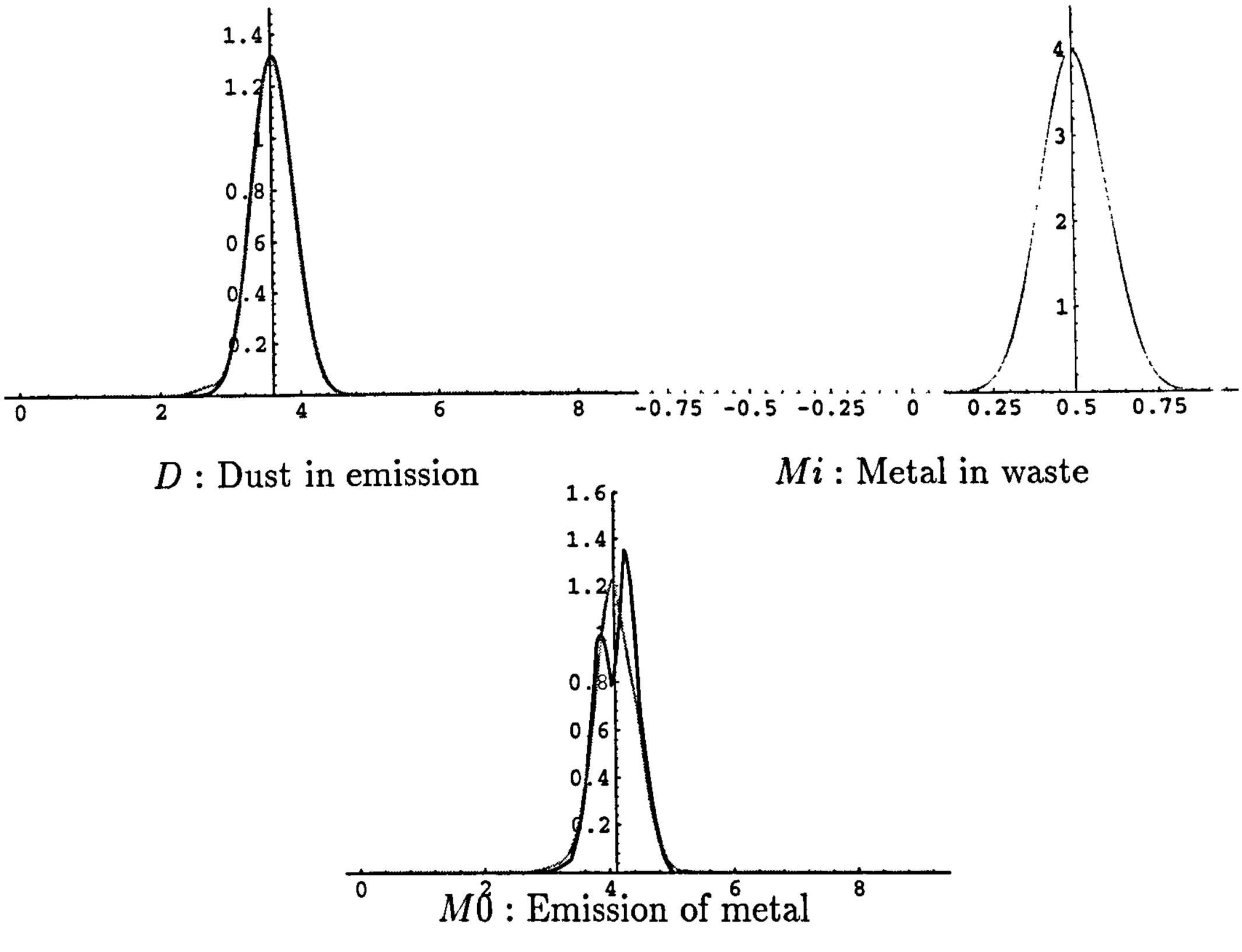



Figure 5.5: Graphs of the exact and spline interpolated marginal distributions of the continuous variables in the waste incinerator problem given Lauritzen's evidence ( $W$ = Industrial, $C$ = -0.9, $L$ = 1.1). Exact marginal distributions are given in grey and spline interpolated ones in black.

| Variable | Spline Interpolated | Exact |
|---|---|---|
| $B$ | (0.000158007, 0.999842) | (0.0122526, 0.987747) |
| $E$ | (0.999652, 0, 0, 0.000347634) | (0.999526, 0, 0, 0.000473552) |
| $F$ | (0.999652, 0.000347634) | (0.999526, 0.000473552) |
| $D$ | (3.61864, 0.0951649) | (3.60766, 0.106162) |
| $Mi$ | (0.499989, 0.00999575) | (0.500000, 0.0100000) |
| $M0$ | (4.13877, 0.107575) | (4.10766, 0.118162) |

Table 5.3: Means, variances and probabilities of the marginal distributions for the waste incinerator problem given Lauritzen's evidence ( $W$ = Industrial, $C$ = -0.9, $L$ = 1.1).

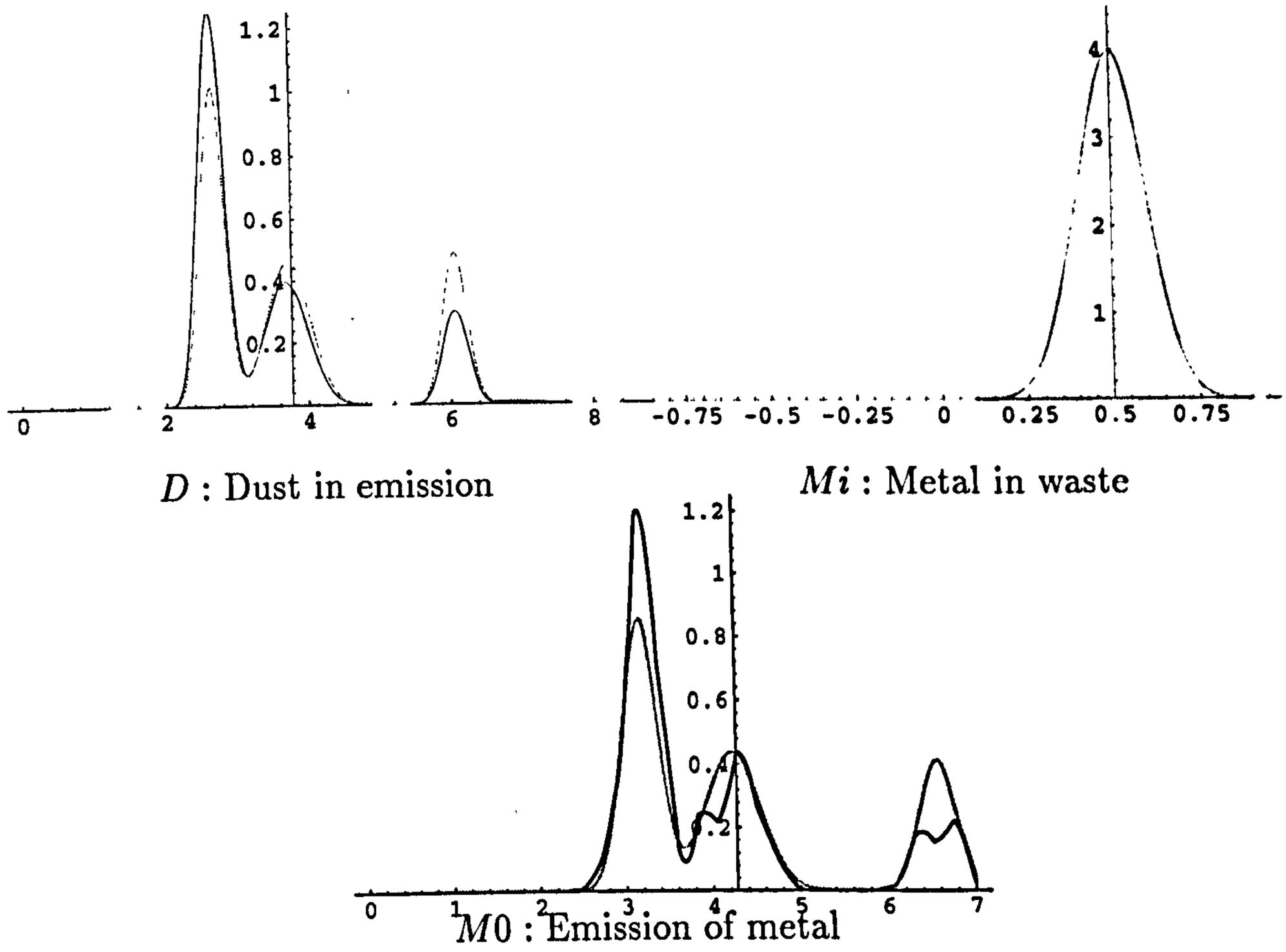



Figure 5.6: Graphs of the exact and spline interpolated marginal distributions of the continuous variables in the waste incinerator problem given Olesen's evidence ( $W$ = Industrial, $C$ = -1.6, $L$ = 0.5). Exact marginal distributions are given in grey and spline interpolated ones in black.

| Variable | Spline Interpolated | Exact |
|---|---|---|
| $B$ | (0.691147, 0.308853) | (0.642429, 0.357571) |
| $E$ | (0.862317, 0, 0, 0.137683) | (0.785814, 0, 0, 0.214186) |
| $F$ | (0.862317, 0.137683) | (0.785814, 0.214186) |
| $D$ | (3.44635, 1.38335) | (3.77449, 1.73627) |
| $Mi$ | (0.499989, 0.00999575) | (0.500000, 0.0100000) |
| $M0$ | (3.96333, 1.36931) | (4.27449, 1.74827) |

Table 5.4: Means, variances and probabilities of the marginal distributions for the waste incinerator problem given Olesen's evidence ( $W$ = Industrial, $C$ = -1.6, $L$ = 0.5).

## 5.11 Conclusions

We have shown how the joint probability density functions of sets of continuous random variables may be sampled over a grid of points and represented by tables of function values and derivative data. We have considered how cubic splines may be used to fit multi-dimensional surfaces to these grids of points and how the approximated functions may then be incorporated into a probabilistic expert system. We have shown how these approximated functions can be propagated through a junction tree to calculate approximations to the joint belief functions for all the cliques and separators in the junction tree. We have also shown how evidence on random variables may be added and how we may determine the marginal distributions of the remaining random variables. We have shown how we may plot the marginal distributions of the continuous random variables, and how we may calculate their means and variances. We have, however, at no stage attempted to pass judgement on the quality of these approximations.

The best way in which to analyse the validity of these approximation methods is to compare the results of their implementation with the exact results. Considering Tables 5.2, 5.3 and 5.4 and Figures 5.4, 5.5 and 5.6 we can see that our approximations are indeed very good. Probabilities, means and variances are all quite close to their desired values and the graphs of the continuous variables manage to capture most of the features of interest. It can be seen that the best approximations occur when the continuous variables are determined by only a small number of continuous parents. This is as would be expected since we need only approximate a space of small dimension. As the dimension of the function space increases more points need to be sampled in order to maintain the quality of the approximation. As was noted earlier we chose our knot sequences using a rough numeric rule. Using a combination of numeric rules or investigating the initial univariate interpolations graphically may have resulted in an improvement of our results. Increasing the number of intervals into which $M0$ was discretised may improve its estimation.

Considering the exact marginal distribution for $D$, given no evidence, we find it is a weighted sum of eight Normal distributions and results in quite a complicated curve. Nevertheless our spline approximation method is able to cope with it admirably. In contrast the exact marginal distribution of $M0$ is also a sum of eight Normal distributions and it too has a curve of comparable complexity. We are, however, unable to form $M0$ with our spline approximation method due to

the excessive number of knots we would require. It is likely that when initial distributions are quite smooth and not too *spiky* good results may be obtained particularly when the number of continuous variables occupying any given clique or separator is not too large. Unfortunately with large numbers of knots and large numbers of continuous variables the number of nodal values required can become unwieldy. It should be recognised, however, that such problems are not confined to this approach - even 20 binary variables in one clique leads to over a million cells being required, large cliques are, in general, to be avoided. In practice we would not use these approximation techniques on a problem which, like the waste incinerator problem, can be calculated exactly.

Further techniques in this genre which may be implemented include cubic spline interpolation without derivative data and spline blended interpolation. This former technique seeks to fit cubic splines to the data by assuming continuity in the first two derivatives and flatness at the external knots. This enables less nodal values to be used to approximate the functions and hence more knots may be sampled with this gain in computer space. A disadvantage of the method, however, is that functions are not consistent under marginalisation though with large numbers of data points these errors will become minimal. The spline blended approach to fitting multi-dimensional surfaces will allow us to consider both exact probability density functions and spline interpolated ones in the same clique or separator. We will investigate this approach in the next chapter.

# Chapter 6

# Hybrid Techniques

## 6.1 Introduction

This chapter describes how *hybrid propagation* schemes may be devised. A hybrid propagation scheme is one which combines elements of two or more different propagation schemes into one overall methodology. The particular type of propagation scheme considered here unites the theory covered in Chapters 4 and 5 and assumes that we have a probabilistic expert system comprising both discrete and continuous random variables. We assume that we are able to represent the discrete random variables exactly through a list of their possible configurations and their associated weights. The continuous random variables are assumed to fall into two distinct categories. The first category are to be represented exactly by their probability density functions. These variables are termed *symbolic*, and their probability density functions will be manipulated using a computer algebra package. The other category of continuous variables are to be represented approximately using *cubic spline interpolation*. This approximation method takes the conditional probability density function of a continuous random variable, confines it to some finite range, and represents it by a list of points and slopes on some lattice within that range. An approximate probability density function may then be determined from this information.

While this methodology is quite general in structure and may be applied to any marked graph where the discrete variables are of finite state space the choice over which continuous variables are symbolic and which are spline interpolated is most important. The reason for this derives from the fact that in order to marginalise a probability density function with respect to a continuous random variable we must integrate with respect to it. This process of integration can prove to be most

troublesome and, except in certain cases, may have no closed form solution. We therefore present a possible, though not unique, setup for the implementation of our methodology. This setup is based on theory relating to the use of conditional Gaussian distributions in expert systems. We then apply our methodology to the waste incinerator problem (Lauritzen, 1992).

## 6.2 Notation

Let $\mathcal{G} = (K, E)$ be an *independence graph* for a set of random variables $X = (X_1, X_2, \ldots, X_k)$, where $K$ is a finite set of *vertices* and the set of *edges* $E$ is a subset of the set $K \times K$ of ordered pairs of distinct vertices. Suppose that a vertex $a \in K$ corresponds to a random variable $X_a$. Let the vertices $K$ be partitioned into three groups $\Delta$, $\Omega$ and $\Gamma$ such that $K = \Delta \cup \Omega \cup \Gamma$. The vertices in $\Delta$ represent *discrete* variables while those in $\Omega \cup \Gamma$ represent *continuous* variables. The vertices $\Omega$ are to be *spline interpolated*, while the vertices $\Gamma$ are to be represented *symbolically*. Let $\Delta$ consist of $p$ vertices, $\Omega$ consist of $q_\Omega$ vertices, $\Gamma$ consist of $q_\Gamma$ vertices, and let $q = q_\Omega + q_\Gamma$ be the total number of continuous vertices. Let $I_\delta$, for $\delta \in \Delta$, denote a particular discrete variable $X_\delta$, let $Z_\omega$, for $\omega \in \Omega$, denote a particular spline interpolated continuous variable $X_\omega$, and let $Y_\gamma$, for $\gamma \in \Gamma$, denote a particular symbolic continuous variable $X_\gamma$. $I_\delta$ takes values in a discrete space $\mathcal{X}_\delta = \mathcal{I}_\delta$, $Z_\omega$ takes real values in a continuous space $\mathcal{X}_\omega = [z_{\omega,0}, z_{\omega,M_\omega}]$, and $Y_\gamma$ takes real values in a continuous space $\mathcal{X}_\gamma = I\!R$. For any $A \subseteq K$ we write $\mathcal{X}_A$ for the space $\times_{a \in A} \mathcal{X}_a$ and in particular:

$$\mathcal{X}_A = \times_{a \in A} \mathcal{X}_a = \mathcal{I}_A \times \mathcal{Z}_A \times \mathcal{Y}_A = (\times_{\delta \in A \cap \Delta} \mathcal{X}_\delta) \times (\times_{\omega \in A \cap \Omega} \mathcal{X}_\omega) \times (\times_{\gamma \in A \cap \Gamma} \mathcal{X}_\gamma)$$

We abbreviate $\mathcal{X}_K$ to $\mathcal{X}$. If $x = (x_a : a \in K)$ then we let $x_A = (x_a : a \in A)$. We may denote a typical element $x$ of the joint state space $\mathcal{X}$ as:

$$x = (x_a)_{a \in K} = (i, z, y) = \{(i_\delta)_{\delta \in \Delta}, (z_\omega)_{\omega \in \Omega}, (y_\gamma)_{\gamma \in \Gamma}\}$$

We will similarly write a typical component $x_A$ of the state space $\mathcal{X}_A$ as:

$$x_A = (x_a)_{a \in A} = (i_A, z_A, y_A) = \{(i_\delta)_{\delta \in A \cap \Delta}, (z_\omega)_{\omega \in A \cap \Omega}, (y_\gamma)_{\gamma \in A \cap \Gamma}\}$$

We will let $pa(a)$ denote the parents of a vertex $a \in K$ corresponding to a random variable $X_a$. The random variables associated with the vertices $pa(a)$ will be denoted $X_{pa(a)}$. These random variables may be decomposed into discrete, spline interpolated, and symbolic random variables as $X_{pa(a)} = (I_{\Delta \cap pa(a)}, Z_{\Omega \cap pa(a)}, Y_{\Gamma \cap pa(a)})$. Where confusion may be avoided we will also use the shorter, but less precise, notation $X_{pa(a)} = (I_{pa(a)}, Z_{pa(a)}, Y_{pa(a)})$ for this decomposition.

## 6.3 Functional Representation

In this section we describe a scheme by which a *potential function* which is a function of both symbolic and spline interpolated continuous variables defined for some configuration of exact discrete variables may be represented.

Consider a universe which, without loss of generality, comprises $D$ discrete random variables $I = (I_1, I_2, \ldots I_D)$, $N$ spline interpolated continuous random variables $Z = (Z_1, Z_2, \ldots, Z_N)$, and $R$ symbolic continuous random variables $Y = (Y_1, Y_2, \ldots, Y_R)$. Realisations of these three groups of variables will be written $i = (i_1, i_2, \ldots, i_D)$, $z = (z_1, z_2, \ldots, z_N)$ and $y = (y_1, y_2, \ldots, y_R)$. We will represent any random variable in $Z$ as $Z_n$ where $n = 1, 2, \ldots, N$. $Z_n$ has realisations $z_n$ and in particular let us suppose that the $M_n + 1$ equally spaced points $z_{n,m_n}$, for $m_n = 0, 1, \ldots, M_n$, are given. We shall let $h_n$ denote the distance between any two consecutive points $z_{n,m_n}$ and $z_{n,m_n+1}$ such that $h_n = z_{n,m_n+1} - z_{n,m_n}$ for $m_n = 0, 1, \ldots, M_n - 1$ and $n = 1, 2, \ldots, N$. We shall let $\mathcal{R}$ denote the following domain:

$$\mathcal{R} = \prod_{n=1}^{N} [z_{n,0}, z_{n,M_n}]$$

We will let $f(i, z, y)$ represent a potential function which is a function of continuous variables $\{Z, Y\}$ for levels $i$ of discrete variables $I$. In order to simplify things we will drop the levels of $I$ from our notation and, where confusion may be avoided, write $f(i, z, y)$ as $f(z, y)$. Thus while the following discussion will relate to a potential function formed only on continuous variables we will assume that this function corresponds to some known cell $i$.

Consider a function $f(z_n)$ formed on the continuous random variable $Z_n$ , for $n = 1, 2, \ldots, N$, and suppose that we have function values and derivatives $\psi(z_{n,l_n})$ for $l_n = 0, 1, \ldots, 2M_n + 1$ where:

$$\psi(z_{n,l_n}) = \begin{cases} \psi(z_{n,2m_n}) & = & f(z_{n,m_n}) & \text{for } l_n = 2m_n \text{ even} \\ \psi(z_{n,2m_n+1}) & = & \frac{\delta f(z_{n,m_n})}{\delta z_n} & \text{for } l_n = 2m_n + 1 \text{ odd} \end{cases}$$

Let us consider the projector associated with interpolation to these function values and derivatives and suppose that we have a well defined interpolation scheme with cardinal functions $\Psi_{n,l_n}(z_n)$ for $l_n = 0, 1, \ldots, 2M_n + 1$ given by Equation 6.1.

$$\Psi_{n,0}(z_n) = \begin{cases} \frac{2}{h_{n,0}^3}\left(z_n - z_{n,0} + \frac{h_{n,0}}{2}\right)(z_n - z_{n,1})^2 & : \quad z_{n,0} \leq z_n \leq z_{n,1} \\ 0 & : \quad z_{n,1} \leq z_n \leq z_{n,M_n} \end{cases}$$

$$\Psi_{n,2m_n}(z_n) = \begin{cases} 0 & : \quad z_{n,0} \leq z_n \leq z_{n,m_n-1} \\ \frac{-2}{h_{n,m_n-1}^3}(z_n - z_{n,m_n-1})^2\left(z_n - z_{n,m_n} - \frac{h_{n,m_n-1}}{2}\right) & : \quad z_{n,m_n-1} \leq z_n \leq z_{n,m_n} \\ \frac{2}{h_{n,m_n}^3}\left(z_n - z_{n,m_n} + \frac{h_{n,m_n}}{2}\right)(z_n - z_{n,m_n+1})^2 & : \quad z_{n,m_n} \leq z_n \leq z_{n,m_n+1} \\ 0 & : \quad z_{n,m_n+1} \leq z_n \leq z_{n,M_n} \end{cases}$$

for $m_n = 1, \ldots, M_n - 1$

$$\Psi_{n,2M_n}(z_n) = \begin{cases} 0 & : \quad z_{n,0} \leq z_n \leq z_{n,M_n-1} \\ \frac{-2}{h_{n,M_n-1}^3}(z_n - z_{n,M_n-1})^2\left(z_n - z_{n,M_n} - \frac{h_{n,M_n-1}}{2}\right) & : \quad z_{n,M_n-1} \leq z_n \leq z_{n,M_n} \end{cases}$$

$$\Psi_{n,1}(z_n) = \begin{cases} \frac{1}{h_{n,0}^2}(z_n - z_{n,1})^2(z_n - z_{n,0}) & : \quad z_{n,0} \leq z_n \leq z_{n,1} \\ 0 & : \quad z_{n,1} \leq z_n \leq z_{n,M_n} \end{cases}$$

$$\Psi_{n,2m_n+1}(z_n) = \begin{cases} 0 & : z_{n,0} \le z_n \le z_{n,m_n-1} \\ \frac{1}{h^2_{n,m_n-1}}(z_n - z_{n,m_n-1})^2(z_n - z_{n,m_n}) & : z_{n,m_n-1} \le z_n \le z_{n,m_n} \\ \frac{1}{h^2_{n,m_n}}(z_n - z_{n,m_n})(z_n - z_{n,m_n+1})^2 & : z_{n,m_n} \le z_n \le z_{n,m_n+1} \\ 0 & : z_{n,m_n+1} \le z_n \le z_{n,M_n} \end{cases}$$

$$\text{for } m_n = 1, \ldots, M_n - 1$$

$$\Psi_{n,2M_n+1}(z_n) = \begin{cases} 0 & : z_{n,0} \le z_n \le z_{n,M_n-1} \\ \frac{1}{h^2_{n,M_n-1}}(z_n - z_{n,M_n-1})^2(z_n - z_{n,M_n}) & : z_{n,M_n-1} \le z_n \le z_{n,M_n} \end{cases} \tag{6.1}$$

There is thus a unique interpolant in the set $\mathcal{H}_{n,0}$ of linear combinations of $\Psi_{n,0}(z_n)$, $\ldots, \Psi_{n,2M_n+1}(z_n)$ given by:

$$(P_{n,0}f)(z_n) = \sum_{l_n=0}^{2M_n+1} \Psi_{n,l_n}(z_n)\psi(z_{n,l_n}) \tag{6.2}$$

Equation 6.2 defines the *map* $P_{n,0}$ from functions $f$ in $\mathcal{L}_{n,0} = \mathcal{C}^1[z_{n,0}, z_{n,M_n}]$ to functions in $\mathcal{H}_{n,0} \subset \mathcal{L}_{n,0}$ and $P_{n,0}$ is a projector in the sense that $P^2_{n,0} = P_{n,0}$.

Now consider extending the definition of $P_{n,0}$ to apply to functions $f(z, y)$, formed on the continuous variables $\{Z, Y\}$, which are continuous on $\mathcal{R} \times I\!R^R$ up until at least the first derivatives. This may be accomplished formally by simply introducing the random variables $\{(Z \backslash Z_n), Y\}$ as "sleeping partners" in Equation 6.2. Thus:

$$(P_n f)(z, y) = \sum_{l_n=0}^{2M_n+1} \Psi_{n,l_n}(z_n)\psi(z_1, \ldots, z_{n,l_n}, \ldots, z_N, y)$$

where:

$$\psi(z_1, \ldots, z_{n,l_n}, \ldots, z_N, y) = \begin{cases} f(z_1, \ldots, z_{n,m_n}, \ldots, z_N, y) & \text{for } l_n = 2m_n \text{ even} \\ \frac{\delta f(z_1, \ldots, z_{n,m_n}, \ldots, z_N, y)}{\delta z_n} & \text{for } l_n = 2m_n + 1 \text{ odd} \end{cases}$$

determines the action of $P_n$ on horizontal lines in one coordinate plane determined by fixed values of $\{(z\backslash z_n), y\}$. This defines the image of $P_n$ as being functions that for each fixed $\{(z\backslash z_n), y\}$ are in $\mathcal{H}_{n,0}$ but for each fixed $z_n$ are merely continuous functions on $[z_{1,0}, z_{1,M_1}] \times \ldots \times [z_{n-1,0}, z_{n-1,M_{n-1}}] \times [z_{n+1,0}, z_{n+1,M_{n+1}}] \times \ldots \times [z_{N,0}, z_{N,M_N}] \times I\!R^R$. By analogy with $P_{n,0}$ it can be seen that $P_n$ is a projector also.

Since the variable $Z_n$ was chosen arbitrarily we have defined a scheme for the univariate interpolation of all functions $f(z, y)$ continuous on $\mathcal{R} \times I\!R^R$ by projectors $P_n$ for $n = 1, 2, \ldots, N$. Without loss of generality consider applying the projectors $P_1, P_2, \ldots, P_N$ in turn to the function $f(z, y)$. It is found that:

$$\begin{aligned} P_1 P_2 \ldots P_N f(z,y) &= \sum_{l_1=0}^{2M_1+1} \Psi_{1,l_1}(z_1) \sum_{l_2=0}^{2M_2+1} \Psi_{2,l_2}(z_2) \ldots \\ &\quad \sum_{l_N=0}^{2M_N+1} \Psi_{N,l_N}(z_N) \psi(z_{1,l_1}, z_{2,l_2}, \ldots, z_{N,l_N}, y) \\ &= \sum_{l_1=0}^{2M_1+1} \sum_{l_2=0}^{2M_2+1} \cdots \sum_{l_N=0}^{2M_N+1} \Psi_{1,l_1}(z_1) \Psi_{2,l_2}(z_2) \ldots \\ &\quad \Psi_{N,l_N}(z_N) \psi(z_{1,l_1}, z_{2,l_2}, \ldots, z_{N,l_N}, y) \end{aligned} \tag{6.3}$$

where:

$$\psi(z_{1,l_1}, z_{2,l_2}, \ldots, z_{N,l_N}, y) = \frac{\delta^{\lambda_1+\lambda_2+\ldots+\lambda_N} f(z_{1,m_1}, z_{2,m_2}, \ldots, z_{N,m_N}, y)}{\delta^{\lambda_1} z_1 \delta^{\lambda_2} z_2 \ldots \delta^{\lambda_N} z_N}$$

and in the above:

$$\lambda_n = \begin{cases} 0 & \text{if } l_n = 2m_n \text{ is even} \\ 1 & \text{if } l_n = 2m_n + 1 \text{ is odd} \end{cases}$$

for $l_n = 0, 1, \ldots, 2M_n + 1$, $m_n = 0, 1, \ldots, M_n$, and $n = 1, 2, \ldots, N$.

Considering the functional form of Equation 6.3 and remembering that the projectors were applied in an arbitrary manner we can see that the projectors $P_1, P_2, \ldots, P_N$ commute. We shall denote $P = P_1 P_2 \ldots P_N$. Using the commutativity of the projectors $P_1, P_2, \ldots, P_N$:

$$
\begin{aligned}
P^2 &= (P_1 P_2 \dots P_N)(P_1 P_2 \dots P_N) \\
&= (P_1 P_2 \dots P_N)(P_N \dots P_2 P_1) \\
&= (P_1 P_2 \dots P_{N-1})(P_N^2)(P_{N-1} \dots P_2 P_1) \\
&= (P_1 P_2 \dots P_{N-1})(P_N)(P_{N-1} \dots P_2 P_1) \\
&= P_N (P_1 P_2 \dots P_{N-1})(P_{N-1} \dots P_2 P_1) \\
&\vdots \\
&= (P_N \dots P_2 P_1) \\
&= P
\end{aligned}
$$

so the map $P$ is itself a projector. Equation 6.3 determines the action of $P$ on the $N-$dimensional plane $\mathcal{R}$ for fixed values of $y$. In such circumstances $Pf$ corresponds to $f$ in terms of function values and first derivatives at the vertices of the lattice $\Lambda$ defined by the points $z_{n,l_n}$ for $l_n = 0, 1, \dots, 2M_n + 1$, $n = 1, 2, \dots N$, and approximates for $f$ elsewhere in $\mathcal{R}$. Letting $y$ be free to roam in $\mathbb{R}^R$ the function values and first derivatives with respect to $z$ of $f$ in $\mathcal{R}$ are functions of $y$. For fixed $z$, therefore, Equation 6.3 defines $Pf$ to be continuous in $y$ and, moreover, if $z$ is fixed at one of the vertices of $\Lambda$ then $Pf \equiv f$. Thus for any function $f$ defined on $\mathcal{R} \times \mathbb{R}^R$ the function $Pf$ is the *interpolant* to $f$ at the $R-$dimensional planes formed at the vertices of the lattice $\Lambda$. Since $Pf$ interpolates to $y$ along planes rather than just a finite point set $P$ may be said to be *transfinite* with respect to $y$, and *finite* with respect to $z$.

## 6.4 Initialisation

We shall, without loss of generality, assume that we have a causal probabilistic network, $\mathcal{G}$, composed of discrete vertices $\Delta$ and continuous vertices $\Omega \cup \Gamma$, where the variables associated with the vertices $\Omega$ are to be *spline interpolated* and those associated with the the vertices $\Gamma$ are to be treated *symbolically*. We shall assume that no vertex $a \in \Omega \cup \Gamma$ which relates to a continuous random variable $X_a$ may have a child which relates to a discrete random variable. We may then define each discrete random variable $I_a$, for $a \in \Delta$, given its parents $I_{pa(a)}$, which are ensured to be discrete, as a table of probabilities $p(i_a; i_{pa(a)})$.

We let a spline interpolated continuous node $Z_a$, for $a \in \Omega$, have parents $X_{pa(a)} = (I_{pa(a)}, Z_{pa(a)}, Y_{pa(a)})$. We assume that $f_{a|pa(a)}$, the conditional distribution

of $Z_a$ given its parents $X_{pa(a)}$, may be constrained to some finite region $[z_{a,0}, z_{a,M_a}]$ such that within this region $f$ is in $\mathcal{C}^1[z_{a,0}, z_{a,M_a}]$ and outside this region $f \equiv 0$.

We let a symbolic continuous node $Y_a$, for $a \in \Gamma$, have parents $X_{pa(a)} = (I_{pa(a)}, Z_{pa(a)}, Y_{pa(a)})$. We shall loosely assume that $f_{a|pa(a)}$, the conditional distribution of $Y_a$ given its parents $X_{pa(a)}$, is "well behaved" in that any potential function formed as a function of a symbolic continuous variable $X_a$ possesses an integral with respect to $x_a$ which is in closed form. Although this criterion might seem vague at first it simply ensures that we are able to marginalise with respect to symbolic continuous variables. Where this criterion fails a symbolic continuous variable will have to be spline approximated instead. A suitable setup will be investigated in more depth later.

The causal probabilistic network $\mathcal{G}$ may be compiled to form a *junction tree*. We will denote this junction tree $\mathcal{T}$ and assume it has vertex-set $\mathcal{C}$ and edge-set $\mathcal{S}$. Associated with any $C \in \mathcal{C}$ is a subset of $K$, which we will denote by $C$ also and term a *clique*. Similarly associated with any $S \in \mathcal{S}$, joining two cliques $C$ and $C'$ is a subset of $K$ which we will denote by $S$ also. We will term $S$ a *separator* and define $S = C \cap C'$. Cliques and separators will collectively be termed *universes*.

Associated with each clique $C \in \mathcal{C}$ is a potential function $a_C(x_C)$ which is a non-negative function defined on the variables in that clique. Similarly associated with each separator $S \in \mathcal{S}$ is a potential function $b_S(x_S)$, which is a non-negative function defined on the variables in that separator. We initially let $a_C(x_C) \equiv 1$, for all $C \in \mathcal{C}$, and $b_S(x_S) \equiv 1$, for all $S \in \mathcal{S}$.

For each variable $X_a$, for $a \in K$, we assign $X_a$ to a clique $C$ in the junction tree. We may only assign $X_a$ to cliques which contain all the variables necessary to define the conditional distribution of $X_a$ given its parents. If more than one suitable clique exists then the choice of which one of them $X_a$ is assigned to is arbitrary. When a variable $X_a$ is assigned to a clique $C$ the potential function, $a_C(x_C)$, on that clique is multiplied by the potential function representing the conditional density of $X_a \mid X_{pa(a)}$. When all the variables associated with $K$ have been assigned $Pf$, the interpolation of the joint probability density function $f$ of $X$, is as follows:

$$Pf_K(x) = \prod_{C \in \mathcal{C}} a_C(x_C) \tag{6.4}$$

since $f$ factorises into the conditional distributions of all the variables given their parents. Since $b_S(x_S) \equiv 1$ we may trivially express Equation 6.4 as:

$$Pf_K(x) = \frac{\prod_{C \in \mathcal{C}} a_C(x_C)}{\prod_{S \in \mathcal{S}} b_S(x_S)} \tag{6.5}$$

Equation 6.5 is termed the *joint system belief.*

## 6.5 Potential Tables

In this section we will consider how we may represent a *potential function* by a *potential table*, and how the initial conditional distributions in our *CPN* may be defined in terms of such tables. Let us consider a universe, $V$, containing the variables $X_V = (I_V, Z_V, Y_V)$ where $I_V$ is a set of discrete variables, $Z_V$ is a set of spline interpolated continuous variables, and $Y_V$ is a set of symbolic continuous variables. It will be convenient to consider two potential functions $\phi(x_V)$ and $\eta(x_V)$, formed on these variables, with associated potential tables $\phi^*(x_V)$ and $\eta^*(x_V)$ respectively. Also, without loss of generality, we shall order the $q_V$ random variables in $Z_V$ as $Z_V = (Z_1, Z_2, \ldots, Z_{q_V})$.

If both $Z_V = \{\emptyset\}$ and $Y_V = \{\emptyset\}$ then we have the *discrete exact case* described in Sections 3.2-3.7. A potential function $\phi(x_V)$ may then be represented exactly by a potential table $\phi^*(x_V)$ in which each cell in the table corresponds to a unique combination of the discrete variables. The function in each cell would be a non-negative number (or possibly the function of some symbolic parameters).

In the case where $Z_V = \{\emptyset\}$ but $Y_V \neq \{\emptyset\}$ then we could represent a potential function $\phi(x_V)$ exactly by a potential table $\phi^*(x_V)$ in which each cell in the table corresponds to a unique combination of the discrete variables. The function in each cell would then be a function of the continuous variables $Y_V$ given the levels of $I_V$. An example of this situation is the *mixed graphical association model* we described in Sections 4.3-4.13.

In the case where $Z_V \neq \{\emptyset\}$ we must not only represent the potential function in each cell of the table but also the first derivatives of the potential function with respect to each of the spline interpolated continuous variables. The cells of the potential table must be formed over both the levels of $I_V$ and the lattice, $\Lambda_V$, over

which the spline interpolated continuous variables are defined. If $Z_V \neq \{\emptyset\}$ and $Y_V = \{\emptyset\}$ then we have the situation described in Chapter 5.

The potential functions $\phi(x_V)$ and $\eta(x_V)$ are represented in Equation 6.6 for particular cells of the potential tables $\phi^*(x_V)$ and $\eta^*(x_V)$ for which the levels of $I_V$ are $i_V$ and the functions and derivatives are shown at some knot $z_V$.

$$\begin{aligned}
\phi^*(x_V) &= \phi^*(i_V, z_V, y_V) \\
&= \left\{\phi(i_V, z_V, y_V), \frac{\delta\phi(i_V, z_V, y_V)}{\delta z_1}, \frac{\delta\phi(i_V, z_V, y_V)}{\delta z_2}, \ldots, \frac{\delta\phi(i_V, z_V, y_V)}{\delta z_{q_V}}\right\} \\
\eta^*(x) &= \eta^*(i_V, z_V, y_V) \\
&= \left\{\eta(i_V, z_V, y_V), \frac{\delta\eta(i_V, z_V, y_V)}{\delta z_1}, \frac{\delta\eta(i_V, z_V, y_V)}{\delta z_2}, \ldots, \frac{\delta\eta(i_V, z_V, y_V)}{\delta z_{q_V}}\right\}
\end{aligned} \tag{6.6}$$

Let us consider how we may define the conditional distributions of the random variables in our system as potential tables. We will assume that the conditional distribution of each variable given its parents is known. As stated earlier discrete variables may have no continuous parents, thus the conditional distribution of a discrete variable $I_a$, for $a \in \Delta$, is:

$$\begin{aligned}
I_a \mid X_{pa(a)} &= I_a \mid I_{pa(a)} \\
&\sim f_{a|pa(a)}(i_{a \cup pa(a)}, \emptyset, \emptyset) \\
&= p(i_a; i_{pa(a)})
\end{aligned}$$

Since $I_a \mid I_{pa(a)}$ is only formed over the discrete variables $I_{a \cup pa(a)}$ its conditional distribution may be represented in potential table form as:

$$\begin{aligned}
\phi^*_{a|pa(a)}(i_{a \cup pa(a)}) &= \left\{f_{a|pa(a)}(i_{a \cup pa(a)}, \emptyset, \emptyset)\right\} \\
&= \left\{p(i_a; i_{pa(a)})\right\}
\end{aligned}$$

The conditional distribution of a spline interpolated continuous variable $Z_a$ given its parents $X_{pa(a)}$, for $a \in \Omega$, may be written:

$$
\begin{aligned}
Z_a \mid X_{pa(a)} &= Z_a \mid \{I_{pa(a)}, Z_{pa(a)}, Y_{pa(a)}\} \\
&\sim f_{a|pa(a)}(i_{pa(a)}, z_{a \cup pa(a)}, y_{pa(a)})
\end{aligned}
$$

Hence the conditional distribution of $Z_a \mid X_{pa(a)}$, for $a \in \Omega$, may be represented in potential table form as:

$$
\begin{aligned}
&\phi^*_{a|pa(a)}(i_{pa(a)}, z_{a \cup pa(a)}, y_{pa(a)}) \\
&= \left\{ f_{a|pa(a)}(i_{pa(a)}, z_{a \cup pa(a)}, y_{pa(a)}), \frac{\delta f_{a|pa(a)}(i_{pa(a)}, z_{a \cup pa(a)}, y_{pa(a)})}{\delta z_a}, \right. \\
&\quad \frac{\delta f_{a|pa(a)}(i_{pa(a)}, z_{a \cup pa(a)}, y_{pa(a)})}{\delta z_{pa^{(1)}(a)}}, \frac{\delta f_{a|pa(a)}(i_{pa(a)}, z_{a \cup pa(a)}, y_{pa(a)})}{\delta z_{pa^{(2)}(a)}}, \\
&\quad \left. \ldots, \frac{\delta f_{a|pa(a)}(i_{pa(a)}, z_{a \cup pa(a)}, y_{pa(a)})}{\delta z_{pa^{(\nu)}(a)}} \right\}
\end{aligned}
$$

where $z_{pa(a)} = (z_{pa^{(1)}(a)}, z_{pa^{(2)}(a)}, \ldots, z_{pa^{(\nu)}(a)})$ denote realisations of the $\nu$ spline interpolated parents $Z_{pa(a)}$ of $Z_a$.

The conditional distribution of a symbolic continuous variable $Y_a$ given its parents $X_{pa(a)}$, for $a \in \Gamma$, may be written:

$$
\begin{aligned}
Y_a \mid X_{pa(a)} &= Y_a \mid \{I_{pa(a)}, Z_{pa(a)}, Y_{pa(a)}\} \\
&\sim f_{a|pa(a)}(i_{pa(a)}, z_{pa(a)}, y_{a \cup pa(a)})
\end{aligned}
$$

The conditional distribution of $Y_a \mid X_{pa(a)}$, for $a \in \Gamma$, may thus be represented in potential table form as:

$$
\begin{aligned}
&\phi^*_{a|pa(a)}(i_{pa(a)}, z_{pa(a)}, y_{a \cup pa(a)}) \\
&= \left\{ f_{a|pa(a)}(i_{pa(a)}, z_{pa(a)}, y_{a \cup pa(a)}), \right. \\
&\quad \frac{\delta f_{a|pa(a)}(i_{pa(a)}, z_{pa(a)}, y_{a \cup pa(a)})}{\delta z_{pa^{(1)}(a)}}, \frac{\delta f_{a|pa(a)}(i_{pa(a)}, z_{pa(a)}, y_{a \cup pa(a)})}{\delta z_{pa^{(2)}(a)}}, \\
&\quad \left. \ldots, \frac{\delta f_{a|pa(a)}(i_{pa(a)}, z_{pa(a)}, y_{a \cup pa(a)})}{\delta z_{pa^{(\nu)}(a)}} \right\}
\end{aligned}
$$

where $z_{pa(a)} = (z_{pa^{(1)}(a)}, z_{pa^{(2)}(a)}, \ldots, z_{pa^{(\nu)}(a)})$ denote realisations of the $\nu$ spline interpolated parents $Z_{pa(a)}$ of $Y_a$.

## 6.6 Basic Operations

We will now consider the basic operations required by a propagation scheme and show how these will affect the potential tables we defined in Section 6.5.

### 6.6.1 Extension

Let $U \subseteq V \subseteq K$ and, without loss of generality, let us label the $q_U$ spline interpolated variables in $Z_U$ as $(Z_1, Z_2, \ldots, Z_{q_U})$ and the $q_{V\setminus U}$ spline interpolated variables in $Z_{V\setminus U}$ as $(Z_{q_U+1}, Z_{q_U+2}, \ldots, Z_{q_V})$ where $q_V = q_U + q_{V\setminus U}$. Let $\phi^*(x_U) = \phi^*(i_U, z_U, y_U)$ be the potential table for a potential function $\phi(x_U) = \phi(i_U, z_U, y_U)$ defined on $\mathcal{X}_U = \mathcal{I}_U \times \mathcal{Z}_U \times \mathcal{Y}_U$, where:

$$\begin{aligned}\phi^*(x_U) &= \phi^*(i_U, z_U, y_U) \\ &= \left\{\phi(i_U, z_U, y_U), \frac{\delta\phi(i_U, z_U, y_U)}{\delta z_1}, \frac{\delta\phi(i_U, z_U, y_U)}{\delta z_2}, \ldots, \frac{\delta\phi(i_U, z_U, y_U)}{\delta z_{q_U}}\right\}\end{aligned}$$

Now suppose that $\phi^*(x_U)$ is extended to $\eta^*(x_V)$ defined on $\mathcal{X}_V = \mathcal{I}_V \times \mathcal{Z}_V \times \mathcal{Y}_V = (\mathcal{I}_U \times \mathcal{I}_{V\setminus U}) \times (\mathcal{Z}_U \times \mathcal{Z}_{V\setminus U}) \times (\mathcal{Y}_U \times \mathcal{Y}_{V\setminus U})$ then if $\mathcal{Z}_{V\setminus U} \in \mathbb{R}^{q_{V\setminus U}}$ the new nodal values defined on $x_V$ will be:

$$\begin{aligned}\eta^*(x_V) &= \eta^*(i_U, i_{V\setminus U}, z_U, z_{V\setminus U}, y_U, y_{V\setminus U}) \\ &= \left\{\phi(i_U, z_U, y_U), \frac{\delta\phi(i_U, z_U, y_U)}{\delta z_1}, \frac{\delta\phi(i_U, z_U, y_U)}{\delta z_2}, \ldots, \frac{\delta\phi(i_U, z_U, y_U)}{\delta z_U}, (0)\right\}\end{aligned}$$

where (0) is a row vector of length $q_{V\setminus U}$ corresponding to the $q_{V\setminus U}$ first derivatives of $\phi(i_U, z_U, y_U)$ with respect to the variables in $Z_{V\setminus U}$.

### 6.6.2 Multiplication

If $\phi^*$ and $\eta^*$ are the potential tables of two potential functions which have been extended to occupy the same space, $\mathcal{X}_V$, and $\phi^*$ and $\eta^*$ are defined as in Equations 6.6, then multiplication is defined, by use of the product rule, as:

$$\begin{aligned}(\phi^* \times \eta^*)(x_V) &= \\ &\left\{\phi(i_V, z_V, y_V) \times \eta(i_V, z_V, y_V), \frac{\delta(\phi(i_V, z_V, y_V) \times \eta(i_V, z_V, y_V))}{z_1},\right. \\ &\left.\frac{\delta(\phi(i_V, z_V, y_V) \times \eta(i_V, z_V, y_V))}{z_2}, \ldots, \frac{\delta(\phi(i_V, z_V, y_V) \times \eta(i_V, z_V, y_V))}{z_{q_V}}\right\}\end{aligned}$$

where, for $a = 1, 2, \ldots, qv$:

$$\frac{\delta(\phi(iv, zv, yv) \times \eta(iv, zv, yv))}{\delta z_a} = \left(\phi(iv, zv, yv) \times \frac{\delta\eta(iv, zv, yv)}{\delta z_a}\right) + \left(\frac{\delta\phi(iv, zv, yv)}{\delta z_a} \times \eta(iv, zv, yv)\right)$$

### 6.6.3 Division

If $\phi^*$ and $\eta^*$ are the potential tables of two potential functions which have been extended to occupy the same space, $\mathcal{X}_V$, and $\phi^*$ and $\eta^*$ are defined as in Equations 6.6, then multiplication is defined, by use of the quotient rule, as:

$$(\phi^*/\eta^*)(x_V) = \begin{cases} \{0, 0, \ldots, 0\} & : \phi(iv, zv, yv) = 0 \\ \left\{\frac{\phi(iv,zv,yv)}{\eta(iv,zv,yv)}, \frac{\delta(\frac{\phi(iv,zv,yv)}{\eta(iv,zv,yv)})}{\delta z_1}, \ldots, \frac{\delta(\frac{\phi(iv,zv,yv)}{\eta(iv,zv,yv)})}{\delta z_{qv}}\right\} & : \eta(iv, zv, yv) \neq 0 \\ \text{undefined} & : \text{otherwise} \end{cases}$$

where, for $a = 1, 2, \ldots, qv$:

$$\frac{\delta(\frac{\phi(iv,zv,yv)}{\eta(iv,zv,yv)})}{\delta z_a} = \frac{\left(\frac{\delta\phi(iv,zv,yv)}{\delta z_a} \times \eta(iv, zv, yv)\right) - \left(\phi(iv, zv, yv) \times \frac{\delta\eta(iv,zv,yv)}{\delta z_a}\right)}{(\eta(iv, zv, yv))^2}$$

### 6.6.4 Marginalisation over Discrete Variables

**Theorem 39** *Suppose that $X_V$ is a set of random variables such that $X_V = (I_V, Z_V, Y_V)$, where $I_V$ is a set of discrete variables, $Z_V$ is a set of spline interpolated continuous variables, and $Y_V$ is a set of symbolic continuous variables. Let $\phi^*(x_V)$ denote the potential table describing the potential function $\phi(x_V)$ as defined in Equation 6.6. Let $U \subseteq V$ and suppose we may partition $I_V$ into two disjoint sets $I_U$ and $I_{V \setminus U}$. Also, let $Z_U = Z_V$ and $Y_U = Y_V$. Suppose that we wish to marginalise $\phi^*(x_V)$ with respect to $I_{V \setminus U}$. The potential table for $X_U = (I_U, Z_V, Y_V)$ formed by this marginalisation is denoted $\eta^*(x_U)$ and the nodal values associated with a particular cell of this table when the levels of $X_U$ are $x_U = (i_U, z_V, y_V)$ are as follows:*

$$\eta^*(x_U) = \left\{ \sum_{i_{V\setminus U}} \phi(i_V, z_V, y_V), \sum_{i_{V\setminus U}} \frac{\delta\phi(i_V, z_V, y_V)}{\delta z_1}, \right.$$
$$\left. \sum_{i_{V\setminus U}} \frac{\delta\phi(i_V, z_V, y_V)}{\delta z_2}, \ldots, \sum_{i_{V\setminus U}} \frac{\delta\phi(i_V, z_V, y_V)}{\delta z_{q_V}} \right\}$$

**Proof.** By analogy with Theorem 37. □

### 6.6.5 Marginalisation over Spline Interpolated Variables

**Theorem 40** *Suppose that $X_V$ is a set of random variables such that $X_V = (I_V, Z_V, Y_V)$, where $I_V$ is a set of discrete variables, $Z_V$ is a set of spline interpolated continuous variables, and $Y_V$ is a set of symbolic continuous variables. Let $\phi^*(x_V)$ denote the potential table describing the potential function $\phi(x_V)$ as defined in Equation 6.6. Let $U \subseteq V$ and suppose we may partition $Z_V$ into two disjoint sets $Z_U$ and $Z_{V\setminus U}$. Also, let $I_U = I_V$ and $Y_U = Y_V$. Without loss of generality let us label the $q_U$ variables in $Z_U$ as $(Z_1, Z_2, \ldots, Z_{q_U})$ and the $q_{V\setminus U}$ variables in $Z_{V\setminus U}$ as $(Z_{q_U+1}, Z_{q_U+2}, \ldots, Z_{q_V})$. Suppose that we wish to marginalise $\phi^*(x_V)$ with respect to $Z_{V\setminus U}$. The potential table for $X_U = (I_V, Z_U, Y_V)$ formed by this marginalisation is denoted $\eta^*(x_U)$ and the nodal values associated with a particular cell of this table when the levels of $X_U$ are $x_U = (i_V, z_U, y_V)$ are as follows:*

$$\eta^*(i_V, z_U, y_V) = \left\{ h \sum_{V\setminus U} \phi(i_V, z_V, y_V), h \sum_{V\setminus U} \frac{\delta\phi(i_V, z_V, y_V)}{\delta z_1}, \right.$$
$$\left. h \sum_{V\setminus U} \frac{\delta\phi(i_V, z_V, y_V)}{\delta z_2}, \ldots, h \sum_{V\setminus U} \frac{\delta\phi(i_V, z_V, y_V)}{\delta z_{q_U}} \right\}$$

*where:*

$$h = \prod_{a\in V\setminus U} h_a$$

*and we assume that a variable $Z_a$, for $a \in V\setminus U$, has an equally spaced knot width $h_a$.*

**Proof.** By analogy with Theorem 38. □

### 6.6.6 Marginalisation over Symbolic Variables

**Theorem 41** *Suppose that $X_V$ is a set of random variables such that $X_V = (I_V, Z_V, Y_V)$, where $I_V$ is a set of discrete variables, $Z_V$ is a set of spline interpolated continuous variables, and $Y_V$ is a set of symbolic continuous variables. Let $\phi^*(x_V)$ denote the potential table describing the potential function $\phi(x_V)$ as defined in Equation 6.6. Let $U \subseteq V$ and suppose we may partition $Y_V$ into two disjoint sets $Y_U$ and $Y_{V\setminus U}$. Also, let $I_U = I_V$ and $Z_U = Z_V$. Without loss of generality let us label the $p_U$ variables in $Y_U$ as $(Y_1, Y_2, \ldots, Y_{p_U})$ and the $p_{V\setminus U}$ variables in $Y_{V\setminus U}$ as $(Y_{p_U+1}, Y_{p_U+2}, \ldots, Y_{p_V})$. Suppose that we wish to marginalise $\phi^*(x_V)$ with respect to $Y_{V\setminus U}$. The potential table for $X_U = (I_V, Z_V, Y_U)$ formed by this marginalisation is denoted $\eta^*(x_U)$ and the nodal values associated with a particular cell of this table when the levels of $X_U$ are $x_U = (i_V, z_V, y_U)$ are as follows:*

$$\eta^*(i_V, z_V, y_U) = \left\{ \int_{y_{V\setminus U}=-\infty}^{y_{V\setminus U}=\infty} \phi(i_V, z_V, y_V)\, \delta y_{V\setminus U}, \int_{y_{V\setminus U}=-\infty}^{y_{V\setminus U}=\infty} \frac{\delta\phi(i_V, z_V, y_V)}{\delta z_1} \delta y_{V\setminus U}, \right.$$
$$\left. \int_{y_{V\setminus U}=-\infty}^{y_{V\setminus U}=\infty} \frac{\delta\phi(i_V, z_V, y_V)}{\delta z_2} \delta y_{V\setminus U}, \ldots, \int_{y_{V\setminus U}=-\infty}^{y_{V\setminus U}=\infty} \frac{\delta\phi(i_V, z_V, y_V)}{\delta z_{q_U}} \delta y_{V\setminus U} \right\}$$

*where, for any function $g(\phi(x_V))$:*

$$\int_{y_{V\setminus U}=-\infty}^{y_{V\setminus U}=\infty} g(\phi(x_V))\, \delta y_{V\setminus U} = \int_{y_{p_U+1}=-\infty}^{y_{p_U+1}=\infty} \int_{y_{p_U+2}=-\infty}^{y_{p_U+2}=\infty} \cdots \int_{y_{p_V}=-\infty}^{y_{p_V}=\infty} g(\phi(x_V))\, \delta y_{p_U+1} \delta y_{p_U+2} \ldots \delta y_{p_V}$$

**Proof.** Using our definition of $\phi^*$ as given in Equation 6.6 and noting that the operation of differentiation is distributive over integration we find that:

$$\eta^*(i_V, z_V, y_U) = \int_{y_{V\setminus U}=-\infty}^{y_{V\setminus U}=\infty} \phi^*(i_V, z_V, y_V)\, \delta y_{V\setminus U}$$

$$= \left\{ \int_{y_{V\setminus U}=-\infty}^{y_{V\setminus U}=\infty} \phi(i_V, z_V, y_V)\, \delta y_{V\setminus U}, \frac{\delta \int_{y_{V\setminus U}=-\infty}^{y_{V\setminus U}=\infty} \phi(i_V, z_V, y_V)\, \delta y_{V\setminus U}}{\delta z_1}, \right.$$
$$\left. \frac{\delta \int_{y_{V\setminus U}=-\infty}^{y_{V\setminus U}=\infty} \phi(i_V, z_V, y_V)\, \delta y_{V\setminus U}}{\delta z_2}, \ldots, \frac{\delta \int_{y_{V\setminus U}=-\infty}^{y_{V\setminus U}=\infty} \phi(i_V, z_V, y_V)\, \delta y_{V\setminus U}}{\delta z_{q_U}} \right\}$$

$$= \left\{ \int_{y_{V\setminus U}=-\infty}^{y_{V\setminus U}=\infty} \phi(i_V, z_V, y_V)\, \delta y_{V\setminus U}, \int_{y_{V\setminus U}=-\infty}^{y_{V\setminus U}=\infty} \frac{\delta\phi(i_V, z_V, y_V)}{\delta z_1} \delta y_{V\setminus U}, \right.$$
$$\left. \int_{y_{V\setminus U}=-\infty}^{y_{V\setminus U}=\infty} \frac{\delta\phi(i_V, z_V, y_V)}{\delta z_2} \delta y_{V\setminus U}, \ldots, \int_{y_{V\setminus U}=-\infty}^{y_{V\setminus U}=\infty} \frac{\delta\phi(i_V, z_V, y_V)}{\delta z_{q_U}} \delta y_{V\setminus U} \right\}$$

□

### 6.6.7 Propagation

We showed in Section 6.4 how we may initialise our system such that Equation 6.5 represents the *joint system belief*. To ensure that the potential function formed on each universe is identically equal to the interpolated joint probability distribution of the variables in that universe we may pass a fully active schedule of *flows*. Each *flow* may be constructed from the basic operations just discussed and is defined as in Definition 23.

## 6.7 A Possible Setup

As stated earlier although our setup is, in theory, quite general the choice over which continuous variables are to be symbolically represented and which are to be spline interpolated is most important and can become quite constrictive. The reason for this derives from the marginalisation process. We will require each marginalisation of a potential function with respect to a symbolic continuous variable to be expressed in terms of integrals which may be solved in closed form. Although many special cases may be devised to suit this general framework we shall consider just one which seeks to unite our symbolic and spline approximated methodologies.

Consider cliques, $U \in \mathcal{C}$, which comprise discrete variables $I_U \subseteq X_\Delta$, spline interpolated variables $Z_U \subseteq X_\Omega$, and symbolic variables $Y_U \subseteq X_\Gamma$. Let us partition the spline interpolated variables $Z_U$ into two disjoint groups $Z_{U_1}$ and $Z_{U_2}$ where $(U_1 \cup U_2) \cap \Omega = (U_1 \cup U_2)$. Let $X_{pa(u)} \subseteq I_{U\setminus u}$ , for $u \in U \cap \Delta$, $X_{pa(u_1)} \subseteq \{I_U, Z_{U\setminus u_1}\}$, for $u_1 \in U_1 \cap \Omega$, $X_{pa(u_2)} \subseteq \{I_U, Z_{U\setminus u_2}, Y_U\}$, for $u_2 \in U_2 \cap \Omega$, and $X_{pa(u)} \subseteq \{I_U, Z_U, Y_{U\setminus u}\}$, for $u \in U \cap \Gamma$.

We shall assume that $I_u \mid I_{pa(u)}$, for $u \in U \cap \Delta$, is multinomially distributed and of known finite state space. We shall assume that $Z_{u_1} \mid \{I_{pa(u_1)}, Z_{pa(u_1)}\}$, for $u_1 \in U_1 \cap \Omega$, may be constrained to some finite region $[z_{u_1,0}, z_{u_1,M_{u_1}}]$ such that

within this region $f_{u_1|pa(u_1)}$, the conditional pdf for $Z_{u_1} \mid \{I_{pa(u_1)}, Z_{pa(u_1)}\}$, is in $C^1[z_{u_1,0}, z_{u_1,M_{u_1}}]$ and outside this region $f_{u_1|pa(u_1)} \equiv 0$.

Similarly we shall assume that $Z_{u_2} \mid \{I_{pa(u_2)}, Z_{pa(u_2)}, Y_{pa(u_2)}\}$, for $u_2 \in U_2 \cap \Omega$, may be constrained to some finite region $[z_{u_2,0}, z_{u_2,M_{u_2}}]$ such that within this region $f_{u_2|pa(u_2)}$, the conditional pdf for $Z_{u_2} \mid \{I_{pa(u_2)}, Z_{pa(u_2)}, Y_{pa(u_2)}\}$, is in $C^1[z_{u_2,0}, z_{u_2,M_{u_2}}]$ and outside this region $f_{u_2|pa(u_2)} \equiv 0$. In addition we shall assume that $f_{u_2|pa(u_2)}$ is conditionally Gaussian with mean $\mu(x_{pa(u_2)}) = \alpha(i_{pa(u_2)}) + \beta_1(i_{pa(u_2)})^T z_{pa(u_2)\cap U_1} + \beta_2(i_{pa(u_2)})^T (z_{pa(u_2)\cap U_2} \;\; y_{pa(u_2)})$ and variance $\sigma^2(i_{pa(u_2)})$.

Finally we shall assume that $Y_u \mid \{I_{pa(u)}, Z_{pa(u)}, Y_{pa(u)}\}$, for $u \in U \cap \Gamma$, is conditionally Gaussian with mean $\mu(x_{pa(u)}) = \alpha(i_{pa(u)}) + \beta_1(i_{pa(u)})^T z_{pa(u)\cap U_1} + \beta_2(i_{pa(u)})^T (z_{pa(u)\cap U_2} \;\; y_{pa(u)})$ and variance $\sigma^2(i_{pa(u)})$.

By construction we have defined a system in which no discrete variable has a continuous parent. Similarly, if a spline interpolated variable is not Conditionally Gaussian in distribution then it has no symbolic parents. What we have therefore is a system comprising both spline interpolated continuous variables of arbitrary distributional type and conditional Gaussian symbolic variables. To ensure the computational feasibility of this setup the conditional Gaussian variables on the interface between spline interpolated variables, which are not conditionally Gaussian, and symbolic variables are forced to be spline interpolated. This allows spline interpolated variables to have descendants which are symbolic. To enable us to demonstrate the effectiveness of this setup we must extend our definition of a CG-distribution to include spline interpolated variables $Z_A = \{Z_{A\setminus B}, Z_B\}$ where the variables $Z_B$ are conditionally Gaussian and the variables $Z_{A\setminus B}$ are not.

**Definition 42 CG-Distribution:** *We term the joint density, $f_A$, of the variables $X_A = \{I_A, Z_A, Y_A\}$ for $B \subseteq A \subseteq K$ a* CG-distribution, *if:*

$$\begin{aligned} f_A(x_A) &= f_A(i_A, z_A, y_A) \\ &= \chi(i_A, z_A)\ exp\left\{g(i_A, z_A) + h(i_A^+)^T y_A^+ - \frac{1}{2} y_A^{+T} J(i_A) y_A^+\right\} \end{aligned} \tag{6.7}$$

*where $\chi(i_A, z_A) \in \{0, 1\}$ is an indicator function equalling one when $f_A$ is positive at $\{i_A, z_A\}$, and zero otherwise, $i_A^+ = \{i_A, z_{A\setminus B}\}$, and $y_A^+ = (z_B^T\, y_A^T)^T$. $J(i_A)$ is assumed to be symmetric and positive definite.*

**Theorem 42** *Equation 6.7 is equivalent to the equation:*

$$\begin{aligned} f_A(x_A) &= f_A(i_A, z_A, y_A) \\ &= \chi(i_A, z_A) p(i_A, z_A) \{det\, \Sigma(i_A)\}^{-1/2} (2\Pi)^{-1/2|\Gamma_A^+|} \\ &\times \quad exp\left\{-\frac{1}{2}(y_A^+ - \xi(i_A^+))^T \Sigma(i_A)^{-1} (y_A^+ - \xi(i_A^+))\right\} \end{aligned} \tag{6.8}$$

*where we define:*

$$\begin{aligned} p(i_A, z_A) &= (2\Pi)^{1/2|\Gamma_A^+|} \{det\, J(i_A)\}^{-1/2}\, exp\left\{g(i_A, z_A) + \frac{1}{2} h(i_A^+)^T J(i_A)^{-1} h(i_A^+)\right\} \\ \xi(i_A^+) &= J(i_A)^{-1} h(i_A^+) \\ \Sigma(i_A) &= J(i_A)^{-1}, \quad and \quad \Sigma \quad is\ positive\ definite. \end{aligned} \tag{6.9}$$

*and* $\Gamma_A^+ = \Omega_B \cup \Gamma_A$, $i_A^+ = \{i_A, z_{A \setminus B}\}$, *and* $y_A^+ = (z_B^T\, y_A^T)^T$.

**Proof.** Rearranging Equations 6.9 we obtain the following definitions for $g(i_A, z_A)$, $h(i_A^+)$, and $J(i_A)$:

$$\begin{aligned} g(i_A, z_A) &= \log p(i_A, z_A) \\ &+ \frac{1}{2}\left\{\log \det \Sigma(i_A)^{-1} - \mid \Gamma_A^+ \mid \log(2\Pi) - \xi(i_A^+)^T \Sigma(i_A)^{-1} \xi(i_A^+)\right\} \\ h(i_A^+) &= \Sigma(i_A)^{-1} \xi(i_A^+) \\ J(i_A) &= \Sigma(i_A)^{-1}, \quad \text{and} \quad J \quad \text{is positive definite.} \end{aligned} \tag{6.10}$$

Now, using Equations 6.7 and 6.10 we may show that:

$$\begin{aligned} f_A(x_A) &= f_A(i_A, z_A, y_A) \\ &= \chi(i_A, z_A) \exp\left\{g(i_A, z_A) + h(i_A^+)^T y_A^+ - \frac{1}{2} y_A^{+T} J(i_A) y_A^+\right\} \\ &= \chi(i_A, z_A) \exp\left\{\log p(i_A, z_A) + \frac{1}{2} \log \det \Sigma(i_A)^{-1} - \frac{1}{2} \mid \Gamma_A^+ \mid \log\ (2\Pi)\right. \\ &- \left.\frac{1}{2} \xi(i_A^+)^T \Sigma(i_A)^{-1} \xi(i_A^+) + \left(\Sigma(i_A)^{-1} \xi(i_A^+)\right)^T y_A^+ - \frac{1}{2} y_A^{+T} \Sigma(i_A)^{-1} y_A^+\right\} \\ &= \chi(i_A, z_A) p(i_A, z_A) \{\det \Sigma(i_A)\}^{-1/2} (2\Pi)^{-1/2|\Gamma_A^+|} \end{aligned}$$

$$
\begin{aligned}
&\times \quad \exp\left\{-\frac{1}{2}\left(\xi(i_A^+)^T\Sigma(i_A)^{-1}\xi(i_A^+)-2\xi(i_A^+)^T\Sigma(i_A)^{-1}y_A^+ + {y_A^+}^T\Sigma(i_A)^{-1}y_A^+\right)\right\} \\
&= \quad \chi(i_A,z_A)p(i_A,z_A)\{\det\Sigma(i_A)\}^{-1/2}(2\Pi)^{-1/2|\Gamma_A^+|} \\
&\times \quad \exp\left\{-\frac{1}{2}(y_A^+-\xi(i_A^+))^T\Sigma(i_A)^{-1}(y_A^+-\xi(i_A^+))\right\}
\end{aligned}
$$

□

We may generalise our definition of a CG-distribution as follows:

**Definition 43 CG-Potential:** *We term the function, $\phi_A$, of the variables $X_A = \{I_A, Z_A, Y_A\}$ for $B \subseteq A \subseteq K$ a* CG-potential, *if:*

$$
\begin{aligned}
\phi_A(x_A) &= \phi_A(i_A,z_A,y_A) \\
&= \chi(i_A,z_A)\ exp\left\{g(i_A,z_A)+h(i_A^+)^Ty_A^+-\frac{1}{2}{y_A^+}^TJ(i_A)y_A^+\right\}
\end{aligned}
\tag{6.11}
$$

*where $\chi(i_A,z_A)\in\{0,1\}$ is an indicator function equalling one when $\phi_A$ is positive at $\{i_A,z_A\}$, and zero otherwise, $i_A^+=\{i_A,z_{A\backslash B}\}$, and $y_A^+=(z_B^Ty_A^T)^T$. We assume that $J(i_A)$ is symmetric and $\phi_A$ is not necessarily a density.*

The conversion formulae of Equations 6.9 and 6.10 continue to apply to CG-potentials when $J(i_A)$, and hence $\Sigma(i_A)$, are positive definite and thus invertible.

**Theorem 43** *The initial conditional distributions of the variables in our system may all be represented by CG-potentials.*

**Proof.** Consider any discrete variable $I_u$ for $u\in U\cap\Delta$ with $X_{pa(u)}\subseteq I_{U\backslash u}$. If we let $A=\{u,pa(u)\}$ then $X_A=I_A$ and, since $I_u \mid I_{pa(u)}$ is multinomially distributed, we may define:

$$f_{u|pa(u)}(i_A)=\chi(i_A)\exp\{g(i_A)\}$$

where $g(i_A)=log\{p(i_a;i_{pa(a)})\}$. Thus $f_{u|pa(u)}$ is a CG-potential for $Z_A=\{\emptyset\}$ and $h(i_A^+)=J(i_A)=0$.

Now, consider any spline interpolated continuous variable $Z_{u_1}$ for $u_1\in U_1\cap\Omega$ and $U_1\subseteq U$. Let $X_{pa(u_1)}\subseteq\{I_U,Z_{U\backslash u_1}\}$ and put $A=\{u_1,pa(u_1)\}$ then we may write $X_A=\{I_A,Z_A\}$. We may thus define:

$$f_{u_1|pa(u_1)}(x_A) = \chi(i_A, z_A) \exp\{g(i_A, z_A)\}$$

which is a CG-potential for $h(i_A^+) = J(i_A) = 0$.

Consider any spline interpolated continuous variable $Z_{u_2}$ for $u_2 \in U_2 \cap \Omega$. Let $U_1 \subseteq U$, $U_2 \subseteq U$, and $(U_1 \cup U_2) \cap \Omega = (U_1 \cup U_2)$. Suppose $X_{pa(u_2)} \subseteq \{I_U, Z_{U \setminus u_2}, Y_U\}$ and put $A = \{u_2, pa(u_2)\}$ then we may write $X_A = \{I_A, Z_A, Y_A\}$. Since $Z_{u_2} \mid X_{pa(u_2)}$ is conditionally Gaussian with mean $\mu(x_{pa(u_2)}) = \alpha(i_{pa(u_2)}) + \beta_1(i_{pa(u_2)})^T z_{pa(u_2)\cap U_1} + \beta_2(i_{pa(u_2)})^T (z_{pa(u_2)\cap U_2} \quad y_{pa(u_2)})$ and variance $\sigma^2(i_{pa(u_2)})$ we may define $f_{u_2|pa(u_2)}(x_A)$ as a CG-potential:

$$f_{u_2|pa(u_2)}(x_A) = \chi(i_A, z_A) \exp\left\{g(i_A^+) + h(i_A^+)^T y_A^+ - \frac{1}{2} y_A^{+T} J(i_A) y_A^+\right\}$$

where $y_A^+ = (z_{u_2}\, z_{pa(u_2)\cap U_2}^T\, y_{pa(u_2)}^T)^T$, $i_A^+ = (i_A, z_{pa(u_2)\cap U_1})$ and:

$$\begin{aligned}
g(i_A^+) &= -\frac{\left[\alpha(i_{pa(u_2)}) + \beta_1(i_{pa(u_2)})^T z_{pa(u_2)\cap U_1}\right]^2}{2\sigma^2(i_{pa(u_2)})} - \frac{1}{2}\log\left\{2\Pi\sigma^2(i_{pa(u_2)})\right\} \\
h(i_A^+) &= \frac{\alpha(i_{pa(u_2)}) + \beta_1(i_{pa(u_2)})^T z_{pa(u_2)\cap U_1}}{\sigma^2(i_{pa(u_2)})} \begin{pmatrix} 1 \\ -\beta_2(i_{pa(u_2)}) \end{pmatrix} \\
J(i_A) &= \frac{1}{\sigma^2(i_{pa(u_2)})} \begin{pmatrix} 1 & -\beta_2(i_{pa(u_2)})^T \\ -\beta_2(i_{pa(u_2)}) & \beta_2(i_{pa(u_2)})\beta_2(i_{pa(u_2)})^T \end{pmatrix}
\end{aligned} \tag{6.12}$$

Consider a symbolic continuous variable $Y_u$ for $u \in U \cap \Gamma$. Let $U_1 \subseteq U$, $U_2 \subseteq U$ and $(U_1 \cup U_2) \cap \Omega = (U_1 \cup U_2)$. Suppose that $X_{pa(u)} \subseteq \{I_U, Z_U, Y_{U \setminus u}\}$ and put $A = \{u, pa(u)\}$ then we may write $X_A = \{I_A, Z_A, Y_A\}$. Since $Y_u \mid X_{pa(u)}$ is conditionally Gaussian with mean $\mu(x_{pa(u)}) = \alpha(i_{pa(u)}) + \beta_1(i_{pa(u)})^T z_{pa(u)\cap U_1} + \beta_2(i_{pa(u)})^T (z_{pa(u)\cap U_2} \quad y_{pa(u)})$ and variance $\sigma^2(i_{pa(u)})$ we may define $f_{u|pa(u)}(x_A)$ as a CG-potential:

$$f_{u|pa(u)}(x_A) = \chi(i_A, z_A) \exp\left\{g(i_A^+) + h(i_A^+)^T y_A^+ - \frac{1}{2} y_A^{+T} J(i_A) y_A^+\right\}$$

where $y_A^+ = (y_u\, z_{pa(u)\cap U_2}^T\, y_{pa(u)}^T)^T$, $i_A^+ = (i_A, z_{pa(u)\cap U_1})$, and:

$$g(i_A^+) = -\frac{\left[\alpha(i_{pa(u)}) + \beta_1(i_{pa(u)})^T z_{pa(u)\cap U_1}\right]^2}{2\sigma^2(i_{pa(u)})} - \frac{1}{2}\log\left\{2\Pi\sigma^2(i_{pa(u)})\right\}$$

$$
\begin{aligned}
h(i_A^+) &= \frac{\alpha(i_{pa(u)}) + \beta_1(i_{pa(u)})^T z_{pa(u)\cap U_1}}{\sigma^2(i_{pa(u)})} \begin{pmatrix} 1 \\ -\beta_2(i_{pa(u)}) \end{pmatrix} \\
J(i_A) &= \frac{1}{\sigma^2(i_{pa(u)})} \begin{pmatrix} 1 & -\beta_2(i_{pa(u)})^T \\ -\beta_2(i_{pa(u)}) & \beta_2(i_{pa(u)})\beta_2(i_{pa(u)})^T \end{pmatrix}
\end{aligned}
\tag{6.13}
$$

□

Our definition of a CG-distribution may be extended to cover summations of CG-distributions. These we call *conditional Gaussian mixture distributions* (or *CGM-distributions* for short).

**Definition 44 CGM-Distribution:** *We term the joint density, $f_A$, of the variables $X_A = \{I_A, Z_A, Y_A\}$ for $B \subseteq A \subseteq K$ a* CGM-distribution, *if:*

$$
\begin{aligned}
f_A(x_A) &= f_A(i_A, z_A, y_A) \\
&= \sum_{l=1}^{L} \chi_l(i_A, z_A)\ exp\left\{ g_l(i_A, z_A) + h_l(i_A^+)^T y_A^+ - \frac{1}{2} y_A^{+T} J_l(i_A) y_A^+ \right\}
\end{aligned}
\tag{6.14}
$$

*where, for $l = 1, 2, \ldots, L$, $\chi_l(i_A, z_A) \in \{0, 1\}$ is an indicator function which controls the inclusion (or exclusion) of the $l-th$ term into the distribution. We let $i_A^+ = \{i_A, z_{A\setminus B}\}$, and $y_A^+ = (z_B^T y_A^T)^T$. We assume that $J_l(i_A)$ is symmetric and positive definite for $l = 1, 2, \ldots, L$.*

**Theorem 44** *Equation 6.14 is equivalent to the equation:*

$$
\begin{aligned}
f_A(x_A) &= f_A(i_A, z_A, y_A) \\
&= \sum_{l=1}^{L} \chi_l(i_A, z_A) p_l(i_A, z_A) \{det\ \Sigma_l(i_A)\}^{-1/2} (2\Pi)^{-1/2|\Gamma_A^+|} \\
&\times\ exp\left\{ -\frac{1}{2} (y_A^+ - \xi_l(i_A^+))^T \Sigma_l(i_A)^{-1} (y_A^+ - \xi_l(i_A^+)) \right\}
\end{aligned}
\tag{6.15}
$$

*where we define:*

$$
\begin{aligned}
p_l(i_A, z_A) &= (2\Pi)^{1/2|\Gamma_A^+|} \{det\ J_l(i_A)\}^{-1/2}\ exp\left\{ g_l(i_A, z_A) + \frac{1}{2} h_l(i_A^+)^T J_l(i_A)^{-1} h_l(i_A^+) \right\} \\
\xi_l(i_A^+) &= J_l(i_A)^{-1} h_l(i_A^+) \\
\Sigma_l(i_A) &= J_l(i_A)^{-1}, \quad and \quad \Sigma \quad is\ positive\ definite.
\end{aligned}
\tag{6.16}
$$

*and* $\Gamma_A^+ = \Omega_B \cup \Gamma_A$, $i_A^+ = \{i_A, z_{A\backslash B}\}$, *and* $y_A^+ = (z_B^T\, y_A^T)^T$.

**Proof.** By analogy with Theorem 42. □

Similarly, our definition of a CG-potential may be extended to cover summations of CG-potentials. These we call *conditional Gaussian mixture potentials* (or *CGM-potentials* for short).

**Definition 45 CGM-Potential:** *We term the function,* $\phi_A$, *of the variables* $X_A = \{I_A, Z_A, Y_A\}$ *for* $B \subseteq A \subseteq K$ *a* CGM-potential, *if:*

$$\begin{aligned}\phi_A(x_A) &= \phi_A(i_A, z_A, y_A) \\ &= \sum_{l=1}^{L} \chi_l(i_A, z_A)\ exp\left\{g_l(i_A, z_A) + h_l(i_A^+)^T y_A^+ - \frac{1}{2} y_A^{+T} J_l(i_A) y_A^+\right\}\end{aligned} \tag{6.17}$$

*where, for* $l = 1, 2, \ldots, L$, $\chi_l(i_A, z_A) \in \{0, 1\}$ *is an indicator function which controls the inclusion (or exclusion) of the* $l-th$ *term into the potential. We let* $i_A^+ = \{i_A, z_{A\backslash B}\}$, *and* $y_A^+ = (z_B^T\, y_A^T)^T$. *We assume that* $J_l(i_A)$ *is symmetric for* $l = 1, 2, \ldots, L$ *and* $\phi_A$ *is not necessarily a density.*

In order that we may construct potential tables for functions of spline interpolated continuous variables we must consider not only the form of their potential functions but also the form of their first derivatives with respect to a spline interpolated continuous variable. This leads us to define a *conditional Gaussian derivative* (or *CG-derivative* for short) as follows:

**Definition 46 CG-Derivative:** *We term the function* $\frac{\delta \phi_A}{\delta z_a}$ *of the variables* $X_A = \{I_A, Z_A, Y_A\}$ *for* $B \subseteq A \subseteq K$ *a* CG-derivative *with respect to* $z_a$, *for* $a \in \Omega$, *if:*

$$\begin{aligned}\frac{\delta \phi_A(x_A)}{\delta z_a} &= \frac{\delta \phi_A(i_A, z_A, y_A)}{\delta z_a} \\ &= \chi(i_A, z_A)\left(g_0(i_A, z_A) + h_0(i_A)^T y_A^+\right) \\ &\times\ exp\left\{g_1(i_A, z_A) + h_1(i_A^+)^T y_A^+ - \frac{1}{2} y_A^{+T} J_1(i_A) y_A^+\right\}\end{aligned} \tag{6.18}$$

*where* $\chi(i_A, z_A) \in \{0,1\}$ *is an indicator function equalling one when* $\phi_A$ *is positive at* $\{i_A, z_A\}$*, and zero otherwise,* $i_A^+ = \{i_A, z_{A\backslash B}\}$*, and* $y_A^+ = (z_B^T y_A^T)^T$*. We assume that* $J_1(i_A)$ *is symmetric.*

We may generalise our definition of a CG-derivative to apply to the first derivative of a CGM-potential. This we call a *conditional Gaussian mixture derivative* (or *CGM-derivative* for short) and define as follows:

**Definition 47 CGM-Derivative:** *We term the function* $\frac{\delta \phi_A}{\delta z_a}$ *of the variables* $X_A = \{I_A, Z_A, Y_A\}$ *for* $B \subseteq A \subseteq K$ *a* CGM-derivative *with respect to* $z_a$*, for* $a \in \Omega$*, if:*

$$\begin{aligned}
\frac{\delta \phi_A(x_A)}{\delta z_a} &= \frac{\delta \phi_A(i_A, z_A, y_A)}{\delta z_a} \\
&= \sum_{l=1}^{L} \chi_l(i_A, z_A) \left( g_{0,l}(i_A, z_A) + h_{0,l}(i_A)^T y_A^+ \right) \\
&\times \ exp \left\{ g_{1,l}(i_A, z_A) + h_{1,l}(i_A^+)^T y_A^+ - \frac{1}{2} y_A^{+T} J_{1,l}(i_A) y_A^+ \right\}
\end{aligned} \tag{6.19}$$

*where, for* $l = 1, 2, \ldots, L$*,* $\chi_l(i_A, z_A) \in \{0,1\}$ *is an indicator function which controls the inclusion (or exclusion) of the* $l-th$ *term into the derivative. We let* $i_A^+ = \{i_A, z_{A\backslash B}\}$*, and* $y_A^+ = (z_B^T y_A^T)^T$*. We assume that* $J_{1,l}(i_A)$ *is symmetric for* $l = 1, 2, \ldots, L$*.*

**Theorem 45** *The first derivatives with respect to* $z_a$*, for* $a \in \Omega$*, of the initial conditional distributions of the variables in our system may all be represented by CG-derivatives.*

**Proof.** Consider any discrete variable $I_u$ for $u \in U \cap \Delta$ with $X_{pa(u)} \subseteq I_{U\backslash u}$. If we let $A = \{u, pa(u)\}$ then $X_A = I_A$ and thus the first derivative of the initial conditional distribution of any discrete variable with respect to $z_a$, for $a \in \Omega$, is zero. This is equivalent to a CG-derivative with $\chi(i_A) \equiv 0$ for all $i_A$.

Consider any spline interpolated continuous variable $Z_{u_1}$ for $u_1 \in U_1 \cap \Omega$ and $U_1 \subseteq U$. Let $X_{pa(u_1)} \subseteq \{I_U, Z_{U\backslash u_1}\}$ and put $A = \{u_1, pa(u_1)\}$ then we may write $X_A = \{I_A, Z_A\}$. The first derivative of $f_{u_1|pa(u_1)}(x_A)$ with respect to $z_a$, for $a \in \Omega$, is zero if $a \notin \Omega_A$ and equal to:

$$\frac{\delta f_{u_1|pa(u_1)}(x_A)}{\delta z_a} = \chi(i_A, z_A) g_0(i_A, z_A) \exp\{g(i_A, z_A)\}$$

otherwise, where $g_0(i_A, z_A) = \frac{\delta g(i_A, z_A)}{\delta z_a}$. Both cases are CG-derivatives.

Consider any spline interpolated continuous variable $Z_{u_2}$ for $u_2 \in U_2 \cap \Omega$. Let $U_1 \subseteq U$, $U_2 \subseteq U$, and $(U_1 \cup U_2) \cap \Omega = (U_1 \cup U_2)$. Suppose $X_{pa(u_2)} \subseteq \{I_U, Z_{U \setminus u_2}, Y_U\}$ and put $A = \{u_2, pa(u_2)\}$ then we may write $X_A = \{I_A, Z_A, Y_A\}$. $Z_{u_2} \mid X_{pa(u_2)}$ is assumed to be conditionally Gaussian with mean $\mu(x_{pa(u_2)}) = \alpha(i_{pa(u_2)}) + \beta_1(i_{pa(u_2)})^T z_{pa(u_2) \cap U_1} + \beta_2(i_{pa(u_2)})^T (z_{pa(u_2) \cap U_2} \; y_{pa(u_2)})$ and variance $\sigma^2(i_{pa(u_2)})$. The first derivative of $f_{u_2|pa(u_2)}(x_A)$ with respect to $z_a$ is zero, and hence a CG- derivative, if $a \notin \Omega_A$. If $a \in \Omega_A$ then the first derivative of $f_{u_2|pa(u_2)}(x_A)$ with respect to $z_a$ is a CG-derivative:

$$\begin{aligned} \frac{\delta f_{u_2|pa(u_2)}(x_A)}{\delta z_a} &= \chi(i_A, z_A) \left( g_0(i_A^+) + h_0(i_A)^T y_A^+ \right) \\ &\times \exp\left\{ g(i_A^+) + h(i_A^+)^T y_A^+ - \frac{1}{2} y_A^{+T} J(i_A) y_A^+ \right\} \end{aligned}$$

where $y_A^+ = (z_{u_2} \, z_{pa(u_2) \cap U_2}^T \, y_{pa(u_2)}^T)^T$, $i_A^+ = (i_A, z_{pa(u_2) \cap U_1})$ and $g(i_A^+)$, $h(i_A^+)$, and $J(i_A)$ are as given in Equations 6.12. If $a = u_2$ then:

$$\begin{aligned} g_0(i_A^+) &= \frac{-2\left(\alpha(i_{pa(u_2)}) + \beta_1(i_{pa(u_2)})^T z_{pa(u_2) \cap U_1}\right)}{\sigma^2(i_{pa(u_2)})} \\ h_0(i_A) &= \frac{2}{\sigma^2(i_{pa(u_2)})} \begin{pmatrix} 1 \\ -\beta_2(i_{pa(u_2)}) \end{pmatrix} \end{aligned}$$

Else if $a \in \Omega_A \setminus u_2$ and $\beta_a(i_{pa(u_2)})$ is the coefficient of $z_a$ in $\mu(x_{pa(u_2)})$ then:

$$\begin{aligned} g_0(i_A^+) &= \frac{2\beta_a(i_{pa(u_2)})\left(\alpha(i_{pa(u_2)}) + \beta_1(i_{pa(u_2)})^T z_{pa(u_2) \cap U_1}\right)}{\sigma^2(i_{pa(u_2)})} \\ h_0(i_A) &= \frac{-2\beta_a(i_{pa(u_2)})}{\sigma^2(i_{pa(u_2)})} \begin{pmatrix} 1 \\ -\beta_2(i_{pa(u_2)}) \end{pmatrix} \end{aligned}$$

Consider any symbolic continuous variable $Y_u$ for $u \in U \cap \Gamma$. Let $U_1 \subseteq U$, $U_2 \subseteq U$, and $(U_1 \cup U_2) \cap \Omega = (U_1 \cup U_2)$. Suppose that $X_{pa(u)} \subseteq \{I_U, Z_{U_2}, Y_{U \setminus u}\}$ and put $A = \{u, pa(u)\}$ then we may write $X_A = \{I_A, Z_A, Y_A\}$. $Y_u \mid X_{pa(u)}$ is assumed to be conditionally Gaussian with mean $\mu(x_{pa(u)}) = \alpha(i_{pa(u)}) + \beta_1(i_{pa(u)})^T z_{pa(u) \cap U_1} +$

$\beta_2(i_{pa(u)})^T(z_{pa(u)\cap U_2} \quad y_{pa(u)})$ and variance $\sigma^2(i_{pa(u)})$. The first derivative of $f_{u|pa(u)}(x_A)$ with respect to $z_a$ is zero, and hence a CG-derivative, if $a \notin \Omega_A$. If $a \in \Omega_A$ then the first derivative of $f_{u|pa(u)}(x_{u|pa(u)})$ with respect to $z_a$ is a CG-derivative:

$$\begin{aligned}\frac{\delta f_{u|pa(u)}(x_A)}{\delta z_a} &= \chi(i_A, z_A)\left(g_0(i_A^+) + h_0(i_A)^T y_A^+\right) \\ &\times \exp\left\{g(i_A^+) + h(i_A^+)^T y_A^+ - \frac{1}{2} y_A^{+T} J(i_A) y_A^+\right\}\end{aligned}$$

where $y_A^+ = (y_u \, z_{pa(u)\cap U_2}^T \, y_{pa(u)}^T)^T$, $i_A^+ = (i_A, z_{pa(u)\cap U_1})$ and $g(i_A^+)$, $h(i_A^+)$, and $J(i_A)$ are as given in Equations 6.13. If $\beta_a(i_{pa(u)})$ denotes the coefficient of $z_a$ in $\mu(x_{pa(u)})$ then:

$$\begin{aligned}g_0(i_A^+) &= \frac{2\beta_a(i_{pa(u)})\left(\alpha(i_{pa(u)}) + \beta_1(i_{pa(u)})^T z_{pa(u)\cap U_1}\right)}{\sigma^2(i_{pa(u)})} \\ h_0(i_A) &= \frac{-2\beta_a(i_{pa(u)})}{\sigma^2(i_{pa(u)})}\begin{pmatrix} 1 \\ -\beta_2(i_{pa(u)}) \end{pmatrix}\end{aligned}$$

□

We must now show that the functional forms of the clique potentials and first derivatives of the clique potentials with respect to spline interpolated variables, retain their structure throughout the process of propagation. If this is the case then we have defined a suitable framework upon which to base our hybrid methods. We will thus consider how the elementary operations required to pass a flow affect CGM-potentials, and CGM-derivatives.

## 6.8 Closure of CGM-Potentials and Derivatives

### 6.8.1 Extension

**Theorem 46** *Let $U \subseteq V \subseteq K$ and $\phi^*(x_U)$, for $x_U = \{i_U, z_U, y_U\}$, be the potential table for a CGM-potential $\phi(x_U)$ with CGM-derivatives $\frac{\delta\phi(x_U)}{\delta z_u}$, for $u \in \Omega_U$, defined on $\mathcal{X}_U = \mathcal{I}_U \times \mathcal{Z}_U \times \mathcal{Y}_U$. Let $\eta^*(x_V) = \eta^*(i_V, z_V, y_V)$, be the extension of $\phi^*(x_U)$ to $V$ defined on $\mathcal{X}_V = \mathcal{I}_V \times \mathcal{Z}_V \times \mathcal{Y}_V = (\mathcal{I}_U \times \mathcal{I}_{V\setminus U}) \times (\mathcal{Z}_U \times \mathcal{Z}_{V\setminus U}) \times (\mathcal{Y}_U \times \mathcal{Y}_{V\setminus U})$. Then $\eta^*(x_V)$ is a potential table for a CGM-potential $\eta(x_V)$ with CGM-derivatives $\frac{\delta\eta(x_V)}{\delta z_v}$, for $v \in \Omega_V$.*

**Proof.** By definition $\eta(x_V) = \eta(i_U, i_{V\setminus U}, z_U, z_{V\setminus U}, y_U, y_{V\setminus U}) = \phi(i_U, z_U, y_U) = \phi(x_U)$ so $\eta(x_V)$ is a CGM-potential. Similarly, by definition for $v \in \Omega_U$, $\frac{\delta\eta(x_V)}{\delta z_v} = \frac{\delta\eta(i_U, i_{V\setminus U}, z_U, z_{V\setminus U}, y_U, y_{V\setminus U})}{\delta z_v} = \frac{\delta\phi(i_U, z_U, y_U)}{\delta z_v} = \frac{\delta\phi(x_U)}{\delta z_v}$ so $\frac{\delta\eta(x_V)}{\delta z_v}$ is a CGM-derivative. If $v \in \Omega_{V\setminus U}$ then $\frac{\delta\phi(x_V)}{\delta z_v}$ is defined to be zero, and is hence a CGM-derivative.

□

### 6.8.2 Multiplication

**Theorem 47** *Let $\phi^*(x_V)$ and $\eta^*(x_V)$, for $x_V = \{i_V, z_V, y_V\}$, be the potential tables for two CGM-potentials, $\phi(x_V)$ and $\eta(x_V)$, which have been extended to occupy the same space $\mathcal{X}_V = \mathcal{I}_V \times \mathcal{Z}_V \times \mathcal{Y}_V$ where $V \subseteq K$. Let $\phi(x_V)$ and $\eta(x_V)$ have CGM-derivatives $\frac{\delta\phi(x_V)}{\delta z_v}$ and $\frac{\delta\eta(x_V)}{\delta z_v}$, for $v \in \Omega_V$. Then the product $(\phi \times \eta)(x_V) = \phi(x_V) \times \eta(x_V)$ is a CGM-potential with CGM-derivatives $\frac{\delta(\phi\times\eta)(x_V)}{\delta z_v}$ for $v \in \Omega_V$.*

**Proof.** First consider the product of the CGM-potentials:

$$
\begin{aligned}
(\phi \times \eta)(x_V) &= \phi(x_V) \times \eta(x_V) \\
&= \sum_{l=1}^{L} \chi_l(i_V, z_V) \exp\Big\{ g_l(i_V, z_V) + h_l(i_V^+)^T y_V^+ - \frac{1}{2} y_V^{+T} J_l(i_V) y_V^+ \Big\} \\
&\times \sum_{m=1}^{M} \chi_m^*(i_V, z_V) \exp\Big\{ g_m^*(i_V, z_V) + h_m^*(i_V^+)^T y_V^+ - \frac{1}{2} y_V^{+T} J_m^*(i_V) y_V^+ \Big\} \\
&= \sum_{l=1}^{L} \sum_{m=1}^{M} \Big( \chi_l(i_V, z_V) \chi_m^*(i_V, z_V) \Big) \exp\Big\{ \Big( g_l(i_V, z_V) + g_m^*(i_V, z_V) \Big) \\
&+ \Big( h_l(i_V^+) + h_m^*(i_V^+) \Big)^T y_V^+ - \frac{1}{2} y_V^{+T} \Big( J_l(i_V) + J_m^*(i_V) \Big) y_V^+ \Big\} \\
&= \sum_{n=1}^{N} \chi_n'(i_V, z_V) \exp\Big\{ g_n'(i_V, z_V) + h_n'(i_V^+)^T y_V^+ - \frac{1}{2} y_V^{+T} J_n'(i_V) y_V^+ \Big\}
\end{aligned}
$$

Thus $(\phi\times\eta)(x_V)$ is a CGM-potential. Now consider the associated first derivatives with respect to $z_v$ for $v \in \Omega_V$:

$$
\begin{aligned}
\frac{\delta(\phi \times \eta)(x_V)}{\delta z_v} &= \left( \phi(x_V) \times \frac{\delta\eta(x_V)}{\delta z_v} \right) + \left( \frac{\delta\phi(x_V)}{\delta z_v} \times \eta(x_V) \right) \\
&= \left( \sum_{l=1}^{L} \chi_l(i_V, z_V) \exp\Big\{ g_l(i_V, z_V) + h_l(i_V^+)^T y_V^+ - \frac{1}{2} y_V^{+T} J_l(i_V) y_V^+ \Big\} \right. \\
&\times \left. \sum_{r=1}^{R} \chi_r^*(i_V, z_V) \Big( g_{0,r}^*(i_V, z_V) + h_{0,r}^*(i_V)^T y_V^+ \Big) \right)
\end{aligned}
$$

$$\times \exp\left\{g^*_{1,r}(iv,zv) + h^*_{1,r}(i^+_V)^T y^+_V - \frac{1}{2} y^{+T}_V J^*_{1,r}(iv) y^+_V\right\}\Big)$$

$$+ \left(\sum_{s=1}^{S} \chi_s(iv,zv)\left(g_{0,s}(iv,zv) + h_{0,s}(iv)^T y^+_V\right)\right.$$

$$\times \exp\left\{g_{1,s}(iv,zv) + h_{1,s}(i^+_V)^T y^+_V - \frac{1}{2} y^{+T}_V J_{1,s}(iv) y^+_V\right\}$$

$$\times \left.\sum_{m=1}^{M} \chi^\dagger_m(iv,zv) \exp\left\{g^\dagger_m(iv,zv) + h^\dagger_m(i^+_V)^T y^+_V - \frac{1}{2} y^{+T}_V J^\dagger_m(iv) y^+_V\right\}\right)$$

$$= \left(\sum_{l=1}^{L}\sum_{r=1}^{R}\left(\chi_l(iv,zv)\chi^*_r(iv,zv)\right)\left(g^*_{0,r}(iv,zv) + h^*_{0,r}(iv)^T y^+_V\right)\right.$$

$$\times \exp\left\{\left(g_l(iv,zv) + g^*_{1,r}(iv,zv)\right) + \left(h_l(i^+_V) + h^*_{1,r}(i^+_V)\right)^T y^+_V\right.$$

$$\left.\left. - \frac{1}{2} y^{+T}_V \left(J_l(iv) + J^*_{1,r}(iv)\right) y^+_V\right\}\right)$$

$$+ \left(\sum_{m=1}^{M}\sum_{s=1}^{S}\left(\chi^\dagger_m(iv,zv)\chi_s(iv,zv)\right)\left(g_{0,s}(iv,zv) + h_{0,s}(iv)^T y^+_V\right)\right.$$

$$\times \exp\left\{\left(g^\dagger_m(iv,zv) + g_{1,s}(iv,zv)\right) + \left(h^\dagger_m(i^+_V) + h_{1,s}(i^+_V)\right)^T y^+_V\right.$$

$$\left.\left. - \frac{1}{2} y^{+T}_V \left(J^\dagger_m(iv) + J_{1,s}(iv)\right) y^+_V\right\}\right)$$

$$= \sum_{n=1}^{N} \chi'_n(iv,zv)\left(g'_{0,n}(iv,zv) + h'_{0,n}(iv)^T y^+_V\right)$$

$$\times \exp\left\{g'_{1,n}(iv,zv) + h'_{1,n}(i^+_V)^T y^+_V - \frac{1}{2} y^{+T}_V J'_{1,n}(iv) y^+_V\right\}$$

Thus $\frac{\delta(\phi\times\eta)(xv)}{\delta z_v}$ is a CGM-derivative, hence the result.

□

While we have shown that the multiplication of two CGM-potentials with CGM-derivatives results in the formation of a third CGM-potential which also has CGM-derivatives we must still be concerned as to how this may affect our underlying spline interpolation scheme. In general the multiplication of two cubic splines will not result in the formation of a third cubic spline. However, as noted in Chapter 5, whenever a potential function needs to be multiplied by another we are essentially multiplying by a new projector acting orthogonally to the first, and thus the spline interpolation scheme is preserved. This point continues to hold in our hybrid system. A similar argument may be applied to the division of two potential functions.

### 6.8.3 Division

**Theorem 48** *Let $\phi^*(x_V)$ and $\eta^*(x_V)$, for $x_V = \{i_V, z_V, y_V\}$, be the potential tables for two CGM-potentials, $\phi(x_V)$ and $\eta(x_V)$, which have been extended to occupy the same space $\mathcal{X}_V = \mathcal{I}_V \times \mathcal{Z}_V \times \mathcal{Y}_V$ where $V \subseteq K$. Let $\phi(x_V)$ and $\eta(x_V)$ have CGM-derivatives $\frac{\delta\phi(x_V)}{\delta z_v}$ and $\frac{\delta\eta(x_V)}{\delta z_v}$, for $v \in \Omega_V$. Suppose that either $\phi(x_V) = 0$, $\eta(x_V) \equiv 1$ or $\phi(x_V) = \eta(x_V) \times \phi'(x_V)$, where $\phi'(x_V)$ is a CGM-potential with CGM-derivatives $\frac{\delta\phi'(x_V)}{\delta z_v}$, for $v \in \Omega_V$. Then the division of $\phi(x_V)$ by $\eta(x_V)$, $(\phi/\eta)(x_V) = \phi(x_V)/\eta(x_V)$, is a CGM-potential with CGM-derivatives $\frac{\delta(\phi/\eta)(x_V)}{\delta z_v}$ for $v \in \Omega_V$.*

**Proof.** If $\phi(x_V) = 0$ then $\frac{\delta\phi(x_V)}{\delta z_v} = 0$, for $v \in \Omega_V$ and, by definition, $(\phi/\eta)(x_V) = 0$ which is a CGM-potential. Hence, for $v \in \Omega_V$, $\frac{\delta(\phi/\eta)(x_V)}{\delta z_v} = 0$ are CGM-derivatives.

If $\eta(x_V) \equiv 1$ then $(\phi/\eta)(x_V) = \phi(x_V)/\eta(x_V) = \phi(x_V)/1 = \phi(x_V)$ but since $\phi(x_V)$ is a CGM-potential then $(\phi/\eta)(x_V)$ is a CGM-potential too. Thus, for $v \in \Omega_V$, $\frac{\delta(\phi/\eta)(x_V)}{\delta z_v} = \frac{\delta\phi(x_V)}{\delta z_v}$ which are CGM-derivatives.

Suppose, however, that $\phi(x_V) = \eta(x_V) \times \phi'(x_V)$ then $(\phi/\eta)(x_V) = \phi(x_V)/\eta(x_V)$ $= (\eta(x_V) \times \phi'(x_V))/\eta(x_V) = \phi'(x_V)$ but since $\phi'(x_V)$ is a CGM-potential then $(\phi/\eta)(x_V)$ is a CGM-potential also. Thus, for $v \in \Omega_V$, $\frac{\delta(\phi/\eta)(x_V)}{\delta z_v} = \frac{\delta\phi'(x_V)}{\delta z_v}$ which are CGM-derivatives.

□

### 6.8.4 Marginalisation over Discrete Variables

**Theorem 49** *Let $U \subseteq V \subseteq K$ and $\phi^*(x_V)$, for $x_V = \{i_U, i_{V\setminus U}, z_V, y_V\}$, be the potential table for a CGM-potential $\phi(x_V)$ with CGM-derivatives $\frac{\delta\phi(x_V)}{\delta z_v}$, for $v \in \Omega_V$, defined on $\mathcal{X}_V = \mathcal{I}_U \times \mathcal{I}_{V\setminus U} \times \mathcal{Z}_V \times \mathcal{Y}_V$. Let $\eta^*(x_U)$ for $x_U = \{i_U, z_V, y_V\}$, be the potential table for $\eta(x_U) = \sum_{i_{V\setminus U}} \phi(x_V)$, the marginal of $\phi(x_V)$ with respect to $I_{V\setminus U}$. Then the marginal, $\eta(x_U)$, will be a CGM-potential with CGM-derivatives $\frac{\delta\eta(x_V)}{\delta z_u}$ for $u \in \Omega_U$.*

**Proof.**

$$\begin{aligned}
\eta(i_U, z_V, y_V) &= \sum_{i_{V\setminus U}} \phi(i_U, i_{V\setminus U}, z_V, y_V) \\
&= \sum_{i_{V\setminus U}} \sum_{l=1}^{L} \chi_l(i_U, i_{V\setminus U}, z_V) \exp\Big\{ g_l(i_U, i_{V\setminus U}, z_V) + h_l(i_U^+, i_{V\setminus U})^T y_V^+ \\
&\qquad - \frac{1}{2} {y_V^+}^T J_l(i_U, i_{V\setminus U}) y_V^+ \Big\}
\end{aligned}$$

$$= \sum_{m=1}^{M} \chi'_m(i_U, z_V) \exp\left\{ g'_m(i_U, z_V) + h'_m(i_U^+)^T y_V^+ - \frac{1}{2} {y_V^+}^T J'_m(i_U) y_V^+ \right\}$$

Thus $\eta(x_U)$ is a CGM-potential. Now, for $u \in \Omega_U$:

$$\begin{aligned}
\frac{\delta \eta(i_U, z_V, y_V)}{\delta z_u} &= \sum_{i_{V \setminus U}} \frac{\delta \phi(i_U, i_{V \setminus U}, z_V, y_V)}{\delta z_u} \\
&= \sum_{i_{V \setminus U}} \sum_{l=1}^{L} \chi_l(i_U, i_{V \setminus U}, z_V) \left( g_{0,l}(i_U, i_{V \setminus U}, z_V) + h_{0,l}(i_U, i_{V \setminus U})^T y_V^+ \right) \\
&\times \exp\left\{ g_{1,l}(i_U, i_{V \setminus U}, z_V) + h_{1,l}(i_U^+, i_{V \setminus U})^T y_V^+ - \frac{1}{2} {y_V^+}^T J_{1,l}(i_U, i_{V \setminus U}) y_V^+ \right\} \\
&= \sum_{m=1}^{M} \chi'_m(i_U, z_V) \left( g'_{0,m}(i_U, z_V) + h'_{0,m}(i_U)^T y_V^+ \right) \\
&\times \exp\left\{ g'_{1,m}(i_U, z_V) + h'_{1,m}(i_U^+)^T y_V^+ - \frac{1}{2} {y_V^+}^T J'_{1,m}(i_U) y_V^+ \right\}
\end{aligned}$$

which are CGM-derivatives, hence the result.

□

### 6.8.5 Marginalisation over Spline Interpolated Variables

**Theorem 50** *Let $U \subseteq V \subseteq K$ and $\phi^*(x_V)$, for $x_V = \{i_V, z_U, z_{V \setminus U}, y_V\}$, be the potential table for a CGM-potential $\phi(x_V)$ with CGM-derivatives $\frac{\delta \phi(x_V)}{\delta z_v}$, for $v \in \Omega_V$, defined on $\mathcal{X}_V = \mathcal{I}_V \times \mathcal{Z}_U \times \mathcal{Z}_{V \setminus U} \times \mathcal{Y}_V$. Let $\eta^*(x_U)$ for $x_U = \{i_V, z_U, y_V\}$, be the potential table for $\eta(x_U) = \int_{z_{V \setminus U}} \phi(x_V) \delta z_{V \setminus U}$, the marginal of $\phi(x_V)$ with respect to $Z_{V \setminus U}$. Then the marginal, $\eta(x_U)$, will be a CGM-potential with CGM-derivatives $\frac{\delta \eta(x_U)}{\delta z_u}$ for $u \in \Omega_U$.*

**Proof.** Let $h_a$ be the knot width of the knot sequence defined for a variable $Z_a$, for $a \in \Omega$, and let $h = \prod_{a \in V \setminus U} h_a$. Let the disjoint sets $V_1$ and $V_2$ partition $Z_V = Z_{V_1 \cup V_2}$ into variables $Z_{V_2}$ which were derived from conditional Gaussian distributions, and $Z_{V_1}$ which were not. Let $U_1 \subseteq V_1$ and $U_2 \subseteq V_2$ and $Z_U = Z_{U_1 \cup U_2}$. Then, by Theorem 40:

$$\begin{aligned}
\eta(i_V, z_U, y_V) &= h \sum_{V \setminus U} \phi(i_V, z_U, z_{V \setminus U}, y_V) \\
&= h \sum_{V \setminus U} \sum_{l=1}^{L} \chi_l(i_V, z_U, z_{V \setminus U}) \\
&\times \exp\left\{ g_l(i_V, z_U, z_{V \setminus U}) + h_l(i_V^+)^T y_V^+ - \frac{1}{2} {y_V^+}^T J_l(i_V) y_V^+ \right\}
\end{aligned} \tag{6.20}$$

Without loss of generality suppose $i_V^+ = \{iv, z_{V_1}\} = \{i_U^+, z_{V_1 \setminus U_1}\}$, where $i_U^+ = \{iv, z_{U_1}\}$, $y_V^{+T} = (z_{V_2 \setminus U_2}^T \, y_U^{+T})^T$, where $y_U^{+T} = (z_{U_2}^T \, y_V^T)^T$ and we may partition $h_l(i_V^+)$ and $J_l(iv)$ as follows:

$$h_l(i_V^+) = \begin{pmatrix} h_l^0(i_U^+, z_{V_1 \setminus U_1}) \\ h_l^1(i_U^+, z_{V_1 \setminus U_1}) \end{pmatrix} \qquad J_l(iv) = \begin{pmatrix} J_l^0(iv) & J_l^1(iv)^T \\ J_l^1(iv) & J_l^2(iv) \end{pmatrix}$$

then putting:

$$\begin{aligned} g_l^*(iv, z_U, z_{V \setminus U}) &= \log(h) + g_l(iv, z_U, z_{V \setminus U}) + h_l^1(i_U^+, z_{V_1 \setminus U_1})^T z_{V_2 \setminus U_2} \\ &- \frac{1}{2} z_{V_2 \setminus U_2}^T J_l^2(iv) z_{V_2 \setminus U_2} \\ h_l^*(i_U^+, z_{V_1 \setminus U_1}) &= h_l^0(i_U^+, z_{V_1 \setminus U_1}) - 2 J_l^1(iv)^T z_{V_2 \setminus U_2} \end{aligned}$$

Equation 6.20 may then be written:

$$\begin{aligned} \eta(iv, z_U, y_V) &= \sum_{V \setminus U} \sum_{l=1}^{L} \chi_l(iv, z_U, z_{V \setminus U}) \\ &\times \exp\left\{ g_l^*(iv, z_U, z_{V \setminus U}) + h_l^*(i_U^+, z_{V_1 \setminus U_1})^T y_U^+ - \frac{1}{2} y_U^{+T} J_l^0(iv) y_U^+ \right\} \\ &= \sum_{m=1}^{M} \chi'_m(iv, z_U) \exp\left\{ g'_m(iv, z_U) + h'_m(i_U^+)^T y_U^+ - \frac{1}{2} y_U^{+T} J'_m(iv) y_U^+ \right\} \end{aligned}$$

which is a CGM-potential. We drop the derivatives $\frac{\delta \eta(iv, z_U, y_V)}{\delta z_u}$, for $u \in \Omega_{V \setminus U}$, since they are zero and not required. The derivatives $\frac{\delta \eta(iv, z_U, y_V)}{\delta z_u}$, for $u \in \Omega_U$ are, by Theorem 40, defined to be:

$$\begin{aligned} \frac{\delta \eta(iv, z_U, y_V)}{\delta z_u} &= h \sum_{V \setminus U} \frac{\delta \phi(iv, z_U, z_{V \setminus U}, y_V)}{\delta z_u} \\ &= h \sum_{V \setminus U} \sum_{l=1}^{L} \chi_l(iv, z_U, z_{V \setminus U}) \left( g_{0,l}(iv, z_U, z_{V \setminus U}) + h_{0,l}(iv)^T y_V^+ \right) \\ &\times \exp\left\{ g_{1,l}(iv, z_U, z_{V \setminus U}) + h_{1,l}(i_V^+)^T y_V^+ - \frac{1}{2} y_V^{+T} J_{1,l}(iv) y_V^+ \right\} \end{aligned} \tag{6.21}$$

Without loss of generality suppose $i_V^+ = \{iv, z_{V_1}\} = \{i_U^+, z_{V_1 \setminus U_1}\}$, where $i_U^+ = \{iv, z_{U_1}\}$, $y_V^{+T} = (y_U^{+T} \, z_{V_2 \setminus U_2}^T)^T$, where $y_U^{+T} = (z_{U_2}^T \, y_V)^T$ and we may partition $h_{0,l}(iv)$, $h_{1,l}(i_V^+)$ and $J_{1,l}(iv)$ as follows:

$$h_{0,l}(iv) = \begin{pmatrix} h_{0,l}^0(iv) \\ h_{0,l}^1(iv) \end{pmatrix} \qquad h_{1,l}(i_V^+) = \begin{pmatrix} h_{1,l}^0(i_U^+, z_{V_1 \setminus U_1}) \\ h_{1,l}^1(i_U^+, z_{V_1 \setminus U_1}) \end{pmatrix}$$

$$J_{1,l}(i_V) = \begin{pmatrix} J^0_{1,l}(i_V) & J^1_{1,l}(i_V)^T \\ J^1_{1,l}(i_V) & J^2_{1,l}(i_V) \end{pmatrix}$$

then putting:

$$\begin{aligned}
g^*_{0,l}(i_V, z_U, z_{V\setminus U}) &= g_{0,l}(i_V, z_U, z_{V\setminus U}) + h^1_{0,l}(i^+_U, z_{V_1\setminus U_1})^T z_{V_2\setminus U_2} \\
g^*_{1,l}(i_V, z_U, z_{V\setminus U}) &= \log(h) + g_{1,l}(i_V, z_U, z_{V\setminus U}) + h^1_{1,l}(i^+_U, z_{V_1\setminus U_1})^T z_{V_2\setminus U_2} \\
&- \frac{1}{2} z^T_{V_2\setminus U_2} J^2_{1,l}(i_V) z_{V_2\setminus U_2} \\
h^*_{1,l}(i^+_U, z_{V_1\setminus U_1}) &= h^0_{1,l}(i^+_U, z_{V_1\setminus U_1}) - 2J^1_{1,l}(i_V)^T z_{V_2\setminus U_2}
\end{aligned}$$

Equation 6.21 may then be written:

$$\begin{aligned}
\frac{\delta\eta(i_V, z_U, y_V)}{\delta z_u} &= \sum_{V\setminus U} \sum_{l=1}^{L} \chi_l(i_V, z_U, z_{V\setminus U}) \left( g^*_{0,l}(i_V, z_U, z_{V\setminus U}) + h^0_{0,l}(i_V)^T y^+_U \right) \\
&\times \exp\left\{ g^*_{1,l}(i_V, z_U, z_{V\setminus U}) + h^0_{1,l}(i^+_U, z_{V_1\setminus U_1})^T y^+_U - \frac{1}{2} y^{+T}_U J^0_{1,l}(i_V) y^+_U \right\} \\
&= \sum_{m=1}^{M} \chi'_m(i_V, z_U) \left( g'_{0,m}(i_V, z_U) + h'_{0,m}(i_V)^T y^+_U \right) \\
&\times \exp\left\{ g'_{1,m}(i_V, z_U) + h'_{1,m}(i^+_U)^T y^+_U - \frac{1}{2} y^{+T}_U J'_{1,m}(i_V) y^+_U \right\}
\end{aligned}$$

which is a CGM-derivative, hence the result.

□

### 6.8.6 Marginalisation over Symbolic Variables

**Theorem 51** *Let $U \subseteq V \subseteq K$ and $\phi^*(x_V)$, for $x_V = \{i_V, z_V, y_U, y_{V\setminus U}\}$, be the potential table for a CGM-potential $\phi(x_V)$ with CGM-derivatives $\frac{\delta\phi(x_V)}{\delta z_v}$, for $v \in \Omega_V$, defined on $\mathcal{X}_V = \mathcal{I}_V \times \mathcal{Z}_V \times \mathcal{Y}_U \times \mathcal{Y}_{V\setminus U}$. Let $\eta^*(x_U)$ for $x_U = \{i_V, z_V, y_U\}$, be the potential table for $\eta(x_U) = \int_{y_{V\setminus U}} \phi(x_V)\delta y_{V\setminus U}$, the marginal of $\phi(x_V)$ with respect to $Y_{V\setminus U}$. Then the marginal, $\eta(x_U)$, will be a CGM-potential with CGM-derivatives $\frac{\delta\eta(x_U)}{\delta z_u}$ for $u \in \Omega_U$.*

**Proof.** Let $\phi(x_V)$ be:

$$\phi(x_V) = \sum_{l=1}^{L} \chi_l(i_V, z_V) \exp\left\{ g_l(i_V, z_V) + h_l(i^+_V)^T y^+_V - \frac{1}{2} y^{+T}_V J_l(i_V) y^+_V \right\} \qquad (6.22)$$

Let the disjoint sets $V_1$ and $V_2$ partition $Z_V = Z_{V_1 \cup V_2}$ into variables $Z_{V_2}$ which were derived from conditional Gaussian distributions, and $Z_{V_1}$ which were not. Without loss of generality suppose that $y_V^+ = (z_{V_2}^T \, y_U^T \, y_{V \setminus U}^T)^T = (y_U^{+T} \, y_{V \setminus U}^T)^T$, where $y_U^+ = (z_{V_2}^T \, y_U^T)^T$. Now suppose we may partition $h_l(i_V^+)$ and $J_l(i_V)$, for $l = 1, 2, \ldots, L$, as:

$$h_l(i_V^+) = \begin{pmatrix} h_l^0(i_V^+) \\ h_l^1(i_V^+) \end{pmatrix} \qquad J_l(i_V) = \begin{pmatrix} J_l^0(i_V) & J_l^1(i_V)^T \\ J_l^1(i_V) & J_l^2(i_V) \end{pmatrix}$$

Then we may express Equation 6.22 as:

$$\begin{aligned}
&\phi(i_V, z_V, y_U, y_{V \setminus U}) \\
&= \sum_{l=1}^{L} \chi_l(i_V, z_V) \exp\Big\{ g_l(i_V, z_V) + h_l^0(i_V^+)^T y_U^+ - \frac{1}{2} y_U^{+T} J_l^0(i_V) y_U^+ \\
&- \frac{1}{2} y_{V \setminus U}^T J_l^2(i_V) y_{V \setminus U} - y_{V \setminus U}^T J_l^1(i_V) y_U^+ + h_l^1(i_V^+)^T y_{V \setminus U} \Big\} \\
&= \sum_{l=1}^{L} \chi_l(i_V, z_V) \exp\Big\{ g_l(i_V, z_V) + h_l^0(i_V^+)^T y_U^+ - \frac{1}{2} y_U^{+T} J_l^0(i_V) y_U^+ \\
&+ \frac{1}{2} y_U^{+T} J_l^1(i_V)^T J_l^2(i_V)^{-1} J_l^1(i_V) y_U^+ - y_U^{+T} J_l^1(i_V)^T J_l^2(i_V)^{-1} h_l^1(i_V^+) \\
&+ \frac{1}{2} h_l^1(i_V^+)^T J_l^2(i_V)^{-1} h_l^1(i_V^+) \\
&- \frac{1}{2} y_{V \setminus U}^T J_l^2(i_V) y_{V \setminus U} - y_{V \setminus U}^T J_l^1(i_V) y_U^+ + h_l^1(i_V^+)^T y_{V \setminus U} \\
&- \frac{1}{2} y_U^{+T} J_l^1(i_V)^T J_l^2(i_V)^{-1} J_l^1(i_V) y_U^+ + y_U^{+T} J_l^1(i_V)^T J_l^2(i_V)^{-1} h_l^1(i_V^+) \\
&- \frac{1}{2} h_l^1(i_V^+)^T J_l^2(i_V)^{-1} h_l^1(i_V^+) \Big\} \\
&= \sum_{l=1}^{L} \chi_l(i_V, z_V) \exp\Big\{ g_l^*(i_V, z_V) + h_l^*(i_V^+)^T y_U^+ - \frac{1}{2} y_U^{+T} J_l^*(i_V) y_U^+ \Big\} \\
&\times \exp\Big\{ -\frac{1}{2} \left( y_{V \setminus U} - \mu(i_V^+, y_U^+) \right)^T J_l^2(i_V) \left( y_{V \setminus U} - \mu(i_V^+, y_U^+) \right) \Big\}
\end{aligned} \tag{6.23}$$

where:

$$\begin{aligned}
g_l^*(i_V, z_V) &= g_l(i_V, z_V) + \frac{1}{2} h_l^1(i_V^+)^T J_l^2(i_V)^{-1} h_l^1(i_V^+) \\
h_l^*(i_V^+) &= h_l^0(i_V^+) - h_l^1(i_V^+)^T J_l^2(i_V)^{-1} J_l^1(i_V) \\
J_l^*(i_V) &= J_l^0(i_V) - J_l^1(i_V)^T J_l^2(i_V)^{-1} J_l^1(i_V) \\
\mu(i_V^+, y_U^+) &= J_l^2(i_V)^{-1} h_l^1(i_V^+) - J_l^2(i_V)^{-1} J_l^1(i_V) y_U^+
\end{aligned}$$

Thus using Equation 6.23:

$$\begin{aligned}\eta(x_U) &= \int_{y_{V\setminus U}=-\infty}^{y_{V\setminus U}=+\infty} \phi(i_V, z_V, y_U, y_{V\setminus U})\, \delta y_{V\setminus U} \\ &= \sum_{l=1}^{L} \chi_l(i_V, z_V) \exp\Big\{ g_l^*(i_V, z_V) + h_l^*(i_V^+)^T y_U^+ - \frac{1}{2} y_U^{+T} J_l^*(i_V) y_U^+ \Big\} \\ &\times \int_{y_{V\setminus U}=-\infty}^{y_{V\setminus U}=+\infty} \exp\Big\{ -\frac{1}{2} \left(y_{V\setminus U} - \mu(i_V^+, y_U^+)\right)^T J_l^2(i_V) \left(y_{V\setminus U} - \mu(i_V^+, y_U^+)\right) \Big\} \delta y_{V\setminus U}\end{aligned} \tag{6.24}$$

But since:

$$\int_{y_A=-\infty}^{y_A=+\infty} \exp\Big\{ -\frac{1}{2} (y_A - \mu_A)^T J (y_A - \mu_A) \Big\} \delta y_A = (2\Pi)^{1/2|\Gamma_A|} (\det J)^{-1/2}$$

Equation 6.24 gives:

$$\eta(x_U) = \sum_{l=1}^{L} \chi_l(i_V, z_V) \exp\Big\{ g_l^{**}(i_V, z_V) + h_l^*(i_V^+)^T y_U^+ - \frac{1}{2} y_U^{+T} J_l^*(i_V) y_U^+ \Big\}$$

where:

$$g_l^{**}(i_V, z_V) = g_l^*(i_V, z_V) + \frac{1}{2} \mid \Gamma_{V\setminus U} \mid \log(2\Pi) - \frac{1}{2} \log\{\det J_l^2(i_V)\}$$

Thus $\eta(x_U)$ is a CGM-potential. Now, let $\frac{\delta\phi(x_V)}{\delta z_v}$, for $v \in \Omega_V$, be:

$$\begin{aligned}\frac{\delta\phi(x_V)}{\delta z_v} &= \sum_{l=1}^{L} \chi_l(i_V, z_V) \left( g_{0,l}(i_V, z_V) + h_{0,l}(i_V)^T y_V^+ \right) \\ &\times \exp\Big\{ g_{1,l}(i_V, z_V) + h_{1,l}(i_V^+)^T y_V^+ - \frac{1}{2} y_V^{+T} J_{1,l}(i_V) y_V^+ \Big\}\end{aligned} \tag{6.25}$$

Without loss of generality suppose that $y_V^+ = (z_{V_2}^T\, y_U^T\, y_{V\setminus U}^T)^T = (y_U^{+T}\, y_{V\setminus U}^T)^T$, where $y_U^+ = (z_{V_2}^T\, y_U^T)^T$. Now suppose we may partition $h_{0,l}(i_V)$, $h_{1,l}(i_V^+)$ and $J_{1,l}(i_V)$, for $l = 1, 2, \ldots, L$, as:

$$h_{0,l}(i_V) = \begin{pmatrix} h_{0,l}^0(i_V) \\ h_{0,l}^1(i_V) \end{pmatrix} \quad h_{1,l}(i_V^+) = \begin{pmatrix} h_{1,l}^0(i_V^+) \\ h_{1,l}^1(i_V^+) \end{pmatrix} \quad J_{1,l}(i_V) = \begin{pmatrix} J_{1,l}^0(i_V) & J_{1,l}^1(i_V)^T \\ J_{1,l}^1(i_V) & J_{1,l}^2(i_V) \end{pmatrix}$$

Then we may express Equation 6.25 as:

$$\begin{aligned}\frac{\delta\phi(i_V, z_V, y_U, y_{V\setminus U})}{\delta z_v} &= \sum_{l=1}^{L} \chi_l(i_V, z_V)\left(g_{0,l}(i_V, z_V) + h^0_{0,l}(i_V)^T y^+_U + h^1_{0,l}(i_V)^T y_{V\setminus U}\right) \\ &\times \exp\left\{g^*_{1,l}(i_V, z_V) + h^*_{1,l}(i^+_V)^T y^+_U - \frac{1}{2} y^{+T}_U J^*_{1,l}(i_V) y^+_U\right\} \\ &\times \exp\left\{-\frac{1}{2}\left(y_{V\setminus U} - \mu(i^+_V, y^+_U)\right)^T J^2_{1,l}(i_V)\left(y_{V\setminus U} - \mu(i^+_V, y^+_U)\right)\right\}\end{aligned} \tag{6.26}$$

where:

$$\begin{aligned} g^*_{1,l}(i_V, z_V) &= g_{1,l}(i_V, z_V) + \frac{1}{2} h^1_{1,l}(i^+_V)^T J^2_{1,l}(i_V)^{-1} h^1_{1,l}(i^+_V) \\ h^*_{1,l}(i^+_V) &= h^0_{1,l}(i^+_V) - h^1_{1,l}(i^+_V)^T J^2_{1,l}(i_V)^{-1} J^1_{1,l}(i_V) \\ J^*_{1,l}(i_V) &= J^0_{1,l}(i_V) - J^1_{1,l}(i_V)^T J^2_{1,l}(i_V)^{-1} J^1_{1,l}(i_V) \\ \mu(i^+_V, y^+_U) &= J^2_{1,l}(i_V)^{-1} h^1_{1,l}(i^+_V) - J^2_{1,l}(i_V)^{-1} J^1_{1,l}(i_V) y^+_U \end{aligned}$$

Thus using Equation 6.26:

$$\begin{aligned}\frac{\delta\eta(x_U)}{\delta z_v} &= \int_{y_{V\setminus U}=-\infty}^{y_{V\setminus U}=+\infty} \frac{\delta\phi(i_V, z_V, y_U, y_{V\setminus U})}{\delta z_v}\, \delta y_{V\setminus U} \\ &= \sum_{l=1}^{L} \chi_l(i_V, z_V)\left(g_{0,l}(i_V, z_V) + h^0_{0,l}(i_V)^T y^+_U\right) \\ &\times \exp\left\{g^*_{1,l}(i_V, z_V) + h^*_{1,l}(i^+_V)^T y^+_U - \frac{1}{2} y^{+T}_U J^*_{1,l}(i_V) y^+_U\right\} \\ &\times \int_{y_{V\setminus U}=-\infty}^{y_{V\setminus U}=+\infty} \exp\left\{-\frac{1}{2}\left(y_{V\setminus U} - \mu(i^+_V, y^+_U)\right)^T J^2_{1,l}(i_V)^{-1}\left(y_{V\setminus U} - \mu(i^+_V, y^+_U)\right)\right\}\delta y_{V\setminus U} \\ &+ \sum_{l=1}^{L} \chi_l(i_V, z_V) \exp\left\{g^*_{1,l}(i_V, z_V) + h^*_{1,l}(i^+_V)^T y^+_U - \frac{1}{2} y^{+T}_U J^*_{1,l}(i_V) y^+_U\right\} \\ &\times \int_{y_{V\setminus U}=-\infty}^{y_{V\setminus U}=+\infty} h^1_{0,l}(i_V)^T y_{V\setminus U} \exp\left\{-\frac{1}{2}\left(y_{V\setminus U} - \mu(i^+_V, y^+_U)\right)^T J^2_{1,l}(i_V)^{-1}\left(y_{V\setminus U} - \mu(i^+_V, y^+_U)\right)\right\}\delta y_{V\setminus U}\end{aligned} \tag{6.27}$$

But since:

$$\int_{y_A=-\infty}^{y_A=+\infty} h^T y_A \exp\left\{-\frac{1}{2}(y_A - \mu_A)^T J (y_A - \mu_A)\right\}\delta y_A = (2\Pi)^{1/2|\Gamma_A|}(\det J)^{-1/2} h^T \mu_A$$

Equation 6.27 gives:

$$\begin{aligned}\frac{\delta\eta(x_U)}{\delta z_v} &= \sum_{l=1}^{L}\chi_l(i_V,z_V)\left(g^*_{0,l}(i_V,z_V)+h^*_{0,l}(i_V)^T y^+_U\right)\\ &\times \exp\left\{g^{**}_{1,l}(i_V,z_V)+h^*_{1,l}(i^+_V)^T y^+_U-\frac{1}{2}{y^+_U}^T J^*_{1,l}(i_V)y^+_U\right\}\end{aligned}$$

where:

$$\begin{aligned}g^*_{0,l}(i_V,z_V) &= g_{0,l}(i_V,z_V)+h^1_{0,l}(i_V)^T J^2_{1,l}(i_V)^{-1}h^1_{1,l}(i^+_V)\\ h^*_{0,l}(i_V) &= h^0_{0,l}(i_V)-J^1_{1,l}(i_V)^T J^2_{1,l}(i_V)^{-1}h^1_{0,l}(i_V)\\ g^{**}_{1,l}(i_V,z_V) &= g^*_{1,l}(i_V,z_V)+\frac{1}{2}\mid\Gamma_{V\setminus U}\mid\log(2\Pi)-\frac{1}{2}\log\{\det J^2_{1,l}(i_V)\}\end{aligned}$$

Thus $\frac{\delta\eta(x_U)}{\delta z_v}$ is a CGM-derivative, for $v\in\Omega_V$, hence the result.

□

### 6.8.7 Propagation

**Theorem 52** *The initial clique and separator potentials and derivatives in our proposed system are CG-potentials and CG-derivatives.*

**Proof.** We showed in Theorem 43 that the initial conditional distribution on each random variable may be expressed as a CG-potential. We also showed in Theorem 45 that these CG-potentials have first derivatives which are CG-derivatives. On initialisation each clique and separator potential is set equal to one, which may be expressed as a CG-potential, and each clique and separator derivative is set equal to zero, which may be expressed as a CG-derivative. The initial conditional distributions of all the variables are then multiplied into one clique each. As a special case of Theorem 47 CG-potentials and CG-derivatives are closed under multiplication hence the initial clique and separator potentials are CG-potentials and CG-derivatives.

□

**Theorem 53** *CGM-potentials and CGM-derivatives are closed under propagation.*

**Proof.** We showed in Theorems 46-51 that CGM-potentials and CGM-derivatives are closed under the operations of extension, multiplication, division, and marginalisation. Since a propagation scheme is only composed of these basic

operations CGM-potentials and CGM-derivatives are closed under propagation also.

□

We have thus created a system in which every clique and separator potential may be expressed as a CGM-potential with CGM-derivatives. We therefore know the symbolic form of every function in the system. This is the desired framework from which to work since we may determine every symbolic operation which will be required to make the framework computationally feasible. In particular we may determine the unique marginalisation rules which will be needed to perform any symbolic marginalisations, and the manipulation rules needed to keep the keep the symbolic functions as simple as possible.

## 6.9 Symbolic Manipulation

In order to perform the required symbolic operations we need to extend the techniques discussed in Sections 4.8 and 4.9. In particular, we are now concerned with both the manipulation of CGM-potentials and CGM-derivatives. Despite our inclusion of spline interpolated variables we may, by Theorem 44, represent a CGM-distribution as:

$$\begin{aligned} f_A(x_A) &= \underset{L}{f_A}(i_A, z_A, y_A) \\ &= \sum_{l=1} \chi_l(i_A, z_A) p_l(i_A, z_A) \{\det \Sigma_l(i_A)\}^{-1/2} (2\Pi)^{-1/2|\Gamma_A^+|} \\ &\times \exp\left\{-\frac{1}{2}(y_A^+ - \xi_l(i_A^+))^T \Sigma_l(i_A)^{-1}(y_A^+ - \xi_l(i_A^+))\right\} \end{aligned} \tag{6.28}$$

This may be rewritten as:

$$f_A(x_A) = \sum_{l=1}^{L} \chi_l(i_A, z_A) \text{ sqrt}(r_l(i_A, z_A)) \exp\{s_l(x_A)\}$$

where:

$$\begin{aligned} r_l(i_A, z_A) &= \frac{(p_l(i_A, z_A))^2}{(2\Pi)^{|\Gamma_A^+|} \det \Sigma_l(i_A)} \\ s_l(x_A) &= -\frac{1}{2}(y_A^+ - \xi_l(i_A^+))^T \Sigma_l(i_A)^{-1}(y_A^+ - \xi_l(i_A^+)) \end{aligned}$$

Similarly, the first derivative of a CGM-distribution with respect to $z_a$, for $a \in \Omega_A$, may be represented by the following:

$$\begin{aligned}
\frac{\delta f_A(x_A)}{\delta z_a} &= \frac{\delta f_A(i_A, z_A, y_A)}{\delta z_a} \\
&= \sum_{l=1}^{L} \chi_l(i_A, z_A) p_l(i_A, z_A) \{\det \Sigma_l(i_A)\}^{-1/2} (2\Pi)^{-1/2|\Gamma_A^+|} \\
&\times \left(g_l^*(i_A, z_A) + h_l^*(i_A)^T y_A\right) \\
&\times \exp\left\{-\frac{1}{2}(y_A^+ - \xi_l(i_A^+))^T \Sigma_l(i_A)^{-1} (y_A^+ - \xi_l(i_A^+))\right\}
\end{aligned}$$

which may be rewritten:

$$\begin{aligned}
\frac{\delta f_A(x_A)}{\delta z_a} &= \sum_{l=1}^{L} \chi_l(i_A, z_A) \left(g_l^*(i_A, z_A) + h_l^*(i_A)^T y_A\right) \\
&\times \operatorname{sqrt}(r_l(i_A, z_A)) \exp\{s_l(x_A)\}
\end{aligned} \tag{6.29}$$

The indicator functions, $\chi_l(i_A, z_A)$, in Equations 6.28 and 6.29 will be superfluous in our computational scheme since functions will only be held if $\chi_l(i_A, z_A) \equiv 1$. We therefore drop these functions. The CGM-distribution in Equation 6.28 may thus be represented by the symbolic form:

$$\sum_{l=1}^{L} \operatorname{sqrt}(f_{1,l}) \exp(f_{2,l}(y_A)) \tag{6.30}$$

Where $f_{1,l}$ is a positive real number, and $f_{2,l}(y_A)$ is a polynomial in the variables $y_A$ where the maximum exponent of any $y_a$, for $a \in A$, in this polynomial is two. We will find it most convenient to expand, and where necessary, cancel any terms within the square root and exponent. Equation 6.30 will also represent our chosen symbolic form for a CGM-potential.

Note that due to the nature of derivatives they are signed. We will therefore represent the first derivative of a CGM-distribution as the sum of signed terms. Taking note of the sign, each term may either take the form of a CG-distribution or be the product of some symbolic variable and a CG-distribution. The CGM-derivative in Equation 6.29 may thus be represented by the symbolic form:

$$+ \sum_{l=1}^{L} \text{sign}_l \ \text{sqrt}(f_{1,l}) \ \exp(f_{2,l}(y_A)) + \sum_{m=1}^{M} \text{sign}_m \ y_{a,m} \ \text{sqrt}(f_{1,m}) \ \exp(f_{2,m}(y_A)) \quad (6.31)$$

Where $f_{1,l}$ and $f_{1,m}$ are positive real numbers, and $f_{2,l}(y_A)$ and $f_{2,m}(y_A)$ are polynomials in the variables $y_A$ where the maximum exponent of any $y_a$, for $a \in A$, in these polynomials is two. The sign operators $\text{sign}_l$ and $\text{sign}_m$ take the values $-1$ and 1 where appropriate. The term $y_{a,m}$ represents some symbolic variable $y_a$, for $a \in A$, required in $m-$th term of the second summation. Again, we will find it most convenient to expand, and where necessary, cancel any terms within the square roots and exponents. Equation 6.31 will also represent our chosen symbolic form for a CGM-derivative.

We may define the initial conditional distributions of the random variables in our setup such that they have potential functions and derivatives which take our chosen symbolic forms. Let $I_u$ be a discrete variable for $u \in U \cap \Delta$ with $X_{pa(u)} \subseteq I_{U \setminus u}$. If we let $A = \{u, pa(u)\}$ then:

$$\begin{aligned} f_{u|pa(u)}(i_A) &= \exp\{g(i_A)\} \\ &= \text{sqrt}\left\{\exp\left(2g(i_A)\right)\right\} \exp\{0\} \end{aligned}$$

Let $Z_{u_1}$ be a spline interpolated continuous variable for $u_1 \in U_1 \cap \Omega$ and $U_1 \subseteq U$. Let $X_{pa(u_1)} \subseteq \{I_U, Z_{U \setminus u_1}\}$ and put $A = \{u_1, pa(u_1)\}$ then:

$$\begin{aligned} f_{u_1|pa(u_1)}(x_A) &= \exp\{g(i_A, z_A)\} \\ &= \text{sqrt}\left\{\exp\left(2g(i_A, z_A)\right)\right\} \exp\{0\} \end{aligned}$$

The first derivative of $f_{u_1|pa(u_1)}(x_A)$ with respect to $z_a$, for $a \in \Omega_A$, is:

$$\begin{aligned} \frac{\delta f_{u_1|pa(u_1)}(x_A)}{\delta z_a} &= g_0(i_A, z_A) \exp\{g(i_A, z_A)\} \\ &= \text{sign}\left\{g_0(i_A, z_A)\right\} \ \text{sqrt}\left\{g_0(i_A, z_A)^2 \exp\left(2g(i_A, z_A)\right)\right\} \exp\{0\} \end{aligned}$$

where $\text{sign}\{r\}$ is 1 if $r \geq 0$, and $-1$ otherwise.

Let $Z_{u_2}$ be a conditionally Gaussian spline interpolated continuous variable for $u_2 \in U_2 \cap \Omega$. Let $U_1 \subseteq U$, $U_2 \subseteq U$, and $(U_1 \cup U_2) \cap \Omega = (U_1 \cup U_2)$. Let

$X_{pa(u_2)} \subseteq \{I_U, Z_{U\setminus u_2}, Y_U\}$ and put $A = \{u_2, pa(u_2)\}$ then:

$$\begin{aligned} f_{u_2|pa(u_2)}(x_A) &= \frac{1}{\sqrt{2\Pi}\sigma(i_{pa(u_2)})} \exp\left\{-\frac{1}{2\sigma^2(i_{pa(u_2)})}\left(z_{u_2} - \mu(x_{pa(u_2)})\right)^2\right\} \\ &= \text{sqrt}\left\{\frac{1}{2\Pi\sigma^2(i_{pa(u_2)})}\right\} \exp\left\{-\frac{1}{2\sigma^2(i_{pa(u_2)})}\left(z_{u_2} - \mu(x_{pa(u_2)})\right)^2\right\} \end{aligned}$$

Let $\mu(x_{pa(u_2)}) = \alpha(i_{pa(u_2)}) + \beta_1(i_{pa(u_2)})^T z_{pa(u_2)\cap U_1} + \beta_2(i_{pa(u_2)})^T(z_{pa(u_2)\cap U_2} \;\; y_{pa(u_2)}) = \alpha(i_{pa(u_2)}, z_{pa(u_2)}) + \sum_{v\in\Gamma_{pa(u_2)}} \beta_v y_v$ then the first derivative of $f_{u_2|pa(u_2)}(x_A)$ with respect to $z_a$, for $a \in \Omega_A$, is:

$$\begin{aligned} \frac{\delta f_{u_2|pa(u_2)}(x_A)}{\delta z_a} &= \frac{\beta\left(z_{u_2} - \mu(x_{pa(u_2)})\right)}{\sqrt{2\Pi}\sigma(i_{pa(u_2)})} \exp\left\{-\frac{1}{2\sigma^2(i_{pa(u_2)})}\left(z_{u_2} - \mu(x_{pa(u_2)})\right)^2\right\} \\ &= \left[\text{sign}\left\{-\beta\,\alpha(i_{pa(u_2)}, z_{pa(u_2)})\right\} \text{ sqrt}\left\{\frac{\beta^2\alpha(i_{pa(u_2)}, z_{pa(u_2)})^2}{2\Pi\sigma^2(i_{pa(u_2)})}\right\}\right. \\ &\quad \left.\times \exp\left\{-\frac{1}{2\sigma^2(i_{pa(u_2)})}\left(z_{u_2} - \mu(x_{pa(u_2)})\right)^2\right\}\right] \\ &\quad + \sum_{v\in\Gamma_{pa(u_2)}} \left[\text{sign}\left\{-\beta\beta_v\right\} \; y_v \text{ sqrt}\left\{\frac{\beta^2\beta_v^2}{2\Pi\sigma^2(i_{pa(u_2)})}\right\}\right. \\ &\quad \left.\times \exp\left\{-\frac{1}{2\sigma^2(i_{pa(u_2)})}\left(z_{u_2} - \mu(x_{pa(u_2)})\right)^2\right\}\right] \end{aligned}$$

where $\beta$ is the coefficient of $z_a$ in $\mu(x_{pa(u_2)})$ if $a \neq u_2$ and $\beta = -1$ if $a = u_2$.

Let $Y_u$ be a conditionally Gaussian symbolic continuous variable for $u \in U \cap \Gamma$. Let $U_1 \subseteq U$, $U_2 \subseteq U$, and $(U_1 \cup U_2) \cap \Omega = (U_1 \cup U_2)$. Let $X_{pa(u)} \subseteq \{I_U, Z_U, Y_{U\setminus u}\}$ and put $A = \{u, pa(u)\}$ then:

$$\begin{aligned} f_{u|pa(u)}(x_A) &= \frac{1}{\sqrt{2\Pi}\sigma(i_{pa(u)})} \exp\left\{-\frac{1}{2\sigma^2(i_{pa(u)})}\left(y_u - \mu(x_{pa(u)})\right)^2\right\} \\ &= \text{sqrt}\left\{\frac{1}{2\Pi\sigma^2(i_{pa(u)})}\right\} \exp\left\{-\frac{1}{2\sigma^2(i_{pa(u)})}\left(y_u - \mu(x_{pa(u)})\right)^2\right\} \end{aligned}$$

Let $\mu(x_{pa(u)}) = \alpha(i_{pa(u)}) + \beta_1(i_{pa(u)})^T z_{pa(u)\cap U_1} + \beta_2(i_{pa(u)})^T(z_{pa(u)\cap U_2} \;\; y_{pa(u)}) = \alpha(i_{pa(u)}, z_{pa(u)}) + \sum_{v\in\Gamma_{pa(u)}} \beta_v y_v$ then the first derivative of $f_{u|pa(u)}(x_A)$ with respect to $z_a$, for $a \in \Omega_A$, is:

$$\frac{\delta f_{u|pa(u)}(x_A)}{\delta z_a} = \frac{\beta\left(y_u - \mu(x_{pa(u)})\right)}{\sqrt{2\Pi}\sigma(i_{pa(u)})} \exp\left\{-\frac{1}{2\sigma^2(i_{pa(u)})}\left(y_u - \mu(x_{pa(u)})\right)^2\right\}$$

$$= \left[\text{sign}\left\{-\beta\,\alpha(i_{pa(u)}, z_{pa(u)})\right\}\ \text{sqrt}\left\{\frac{\beta^2\alpha(i_{pa(u)}, z_{pa(u)})^2}{2\Pi\sigma^2(i_{pa(u)})}\right\}\right.$$
$$\left.\times\ \exp\left\{-\frac{1}{2\sigma^2(i_{pa(u)})}\left(y_u - \mu(x_{pa(u)})\right)^2\right\}\right]$$
$$+ \sum_{v\in\Gamma_{pa(u)}}\left[\text{sign}\left\{-\beta\beta_v\right\}\ y_v\ \text{sqrt}\left\{\frac{\beta^2\beta_v^2}{2\Pi\sigma^2(i_{pa(u)})}\right\}\right.$$
$$\left.\times\ \exp\left\{-\frac{1}{2\sigma^2(i_{pa(u)})}\left(y_u - \mu(x_{pa(u)})\right)^2\right\}\right]$$

where $\beta$ is the coefficient of $z_a$ in $\mu(x_{pa(u)})$.

A slight redefinition of the simplification rules discussed in Section 4.9 will be required to allow for the possible inclusion of a sign operator and a symbolic variable in the product forming the basic symbolic form. For the sake of brevity we will not include those details here. It should be noted, however, that one additional integration function will also be required. This is a specific case of the second integration rule required for Theorem 51 and may be defined, for $c < 0$, as follows:

$$\int_{x=-\infty}^{x=+\infty}\sqrt{k}x\,\text{Exp}\left\{a + bx + cx^2\right\}\delta x = \int_{x=-\infty}^{x=+\infty}\sqrt{k}x\,\text{Exp}\left\{\left(a - \frac{b^2}{4c}\right) + c\left(x + \frac{b}{2c}\right)^2\right\}\delta x$$
$$= \sqrt{\frac{-k\Pi}{c}}\,\text{Exp}\left\{a - \frac{b^2}{4c}\right\}$$
$$\times \int_{x=-\infty}^{x=+\infty}\frac{x}{\sqrt{\frac{-\Pi}{c}}}\text{Exp}\left\{c\left(x + \frac{b}{2c}\right)^2\right\}\delta x$$
$$= -b\sqrt{\frac{-k\Pi}{4c^3}}\,\text{Exp}\left\{a - \frac{b^2}{4c}\right\}$$

## 6.10 Adding Evidence

In this section we will consider the entry of a set of evidence $\mathcal{E}$ onto the random variables in a hybrid system. A set of evidence may be entered into the system by entering the evidence on each individual random variable in turn. We shall assume that evidence is presented in the form that $X_a = e$ for some $a \in K$. In each case the evidence should be entered into every universe $V$ containing the random variable $X_a$. We shall suppose that, for a universe $V$ and $x_V = (i_V, z_V, y_V)$, we have the potential table $\phi^*(x_V)$ given in Equation 6.6 and, without loss of generality, let

$I_V = (I_1, I_2, \ldots, I_{pV})$, $Z_V = (Z_1, Z_2, \ldots, Z_{qV})$, and $Y_V = (Y_1, Y_2, \ldots, Y_{rV})$. Once a set of evidence has been entered into a system a propagation schedule should be passed in order to take account of that evidence and make the potentials consistent. Normalisation will ensure that we obtain the updated joint system belief.

### 6.10.1 Discrete Evidence

The updated potential table $\phi^*_{\mathcal{E}}(x_{V|I_v=e})$ once the evidence $\mathcal{E} : I_v = e$, for $v \in \Delta_V$, has been entered is as follows:

$$\begin{aligned}
\phi^*_{\mathcal{E}}\left(x_{V|I_v=e}\right) &= \phi^*_{\mathcal{E}}\left(i_1, \ldots, i_{v-1}, i_{v+1}, \ldots, i_{pV}, z_V, y_V\right) \\
&= \left\{\phi\left(i_1, \ldots, i_{v-1}, e, i_{v+1}, \ldots, i_{pV}, z_V, y_V\right),\right. \\
&\quad \frac{\delta\phi\left(i_1, \ldots, i_{v-1}, e, i_{v+1}, \ldots, i_{pV}, z_V, y_V\right)}{\delta z_1}, \ldots, \\
&\quad \left.\frac{\delta\phi\left(i_1, \ldots, i_{v-1}, e, i_{v+1}, \ldots, i_{pV}, z_V, y_V\right)}{\delta z_{qV}}\right\}
\end{aligned} \tag{6.32}$$

### 6.10.2 Spline Interpolated Continuous Evidence

Let us assume that the spline interpolated continuous variable $X_v$, for $v \in \Omega_V$, has a knot $e$. Then the updated potential table $\phi^*_{\mathcal{E}}(x_{V|Z_v=e})$ once the evidence $\mathcal{E} : Z_v = e$, for $v \in \Omega_V$, has been entered is as follows:

$$\begin{aligned}
\phi^*_{\mathcal{E}}\left(x_{V|Z_v=e}\right) &= \phi^*_{\mathcal{E}}\left(i_V, z_1, \ldots, z_{v-1}, z_{v+1}, \ldots, z_{qV}, y_V\right) \\
&= \left\{\phi\left(i_V, z_1, \ldots, z_{v-1}, e, z_{v+1}, \ldots, z_{qV}, y_V\right),\right. \\
&\quad \frac{\delta\phi\left(i_V, z_1, \ldots, z_{v-1}, e, z_{v+1}, \ldots, z_{qV}, y_V\right)}{\delta z_1}, \ldots, \\
&\quad \frac{\delta\phi\left(i_V, z_1, \ldots, z_{v-1}, e, z_{v+1}, \ldots, z_{qV}, y_V\right)}{\delta z_{v-1}}, \\
&\quad \frac{\delta\phi\left(i_V, z_1, \ldots, z_{v-1}, e, z_{v+1}, \ldots, z_{qV}, y_V\right)}{\delta z_{v+1}}, \ldots, \\
&\quad \left.\frac{\delta\phi\left(i_V, z_1, \ldots, z_{v-1}, e, z_{v+1}, \ldots, z_{qV}, y_V\right)}{\delta z_{qV}}\right\}
\end{aligned} \tag{6.33}$$

If, however, the spline interpolated continuous variable $X_v$, for $v \in \Omega_V$, has knots $z_{v,m_v} \leq e \leq z_{v,m_v+1}$. Then the updated potential table $\phi^*_{\mathcal{E}}(x_{V|Z_v=e})$ once the evidence $\mathcal{E} : Z_v = e$, for $v \in \Omega_V$, has been entered is as given in Equation 6.33 with:

$$\begin{aligned}
&\phi(i_V, z_1, \ldots, z_{v-1}, e, z_{v+1}, \ldots, z_{q_V}, y_V) \;=\; \\
&\qquad \left(\frac{2\left(\phi^*(z_{v,m_v}) + \phi^*(z_{v,m_v+1})\right)}{h_v^3} + \frac{\left(\phi'_v(z_{v,m_v}) + \phi'_v(z_{v,m_v+1})\right)}{h_v^2}\right)(e - z_{v,m_v})^3 \\
&\qquad + \left(\frac{3\left(\phi^*(z_{v,m_v+1}) - \phi^*(z_{v,m_v})\right)}{h_v^2} - \frac{\left(2\phi'_v(z_{v,m_v}) + \phi'_v(z_{v,m_v+1})\right)}{h_v}\right)(e - z_{v,m_v})^2 \\
&\qquad + \phi'_v(z_{v,m_v})\,(e - z_{v,m_v}) + \phi^*(z_{v,m_v})
\end{aligned}$$

$$\begin{aligned}
&\frac{\delta\phi(i_V, z_1, \ldots, z_{v-1}, e, z_{v+1}, \ldots, z_{q_V}, y_V)}{\delta z_n} \;= \\
&\qquad \phi'_n(z_{v,m_v}) + \left(\frac{\phi'_n(z_{v,m_v+1}) - \phi'_n(z_{v,m_v})}{h_v}\right)(e - z_{v,m_v})
\end{aligned}$$

$$\text{for } n = 1, \ldots, v-1, v+1, \ldots, q_V$$

and:

$$\begin{aligned}
h_v &= (z_{v,m_v+1} - z_{v,m_v}) \\
\phi^*(z_{v,m_v}) &= \phi(i_V, z_1, \ldots, z_{v-1}, z_{v,m_v}, z_{v+1}, \ldots, z_{q_V}, y_V) \\
\phi^*(z_{v,m_v+1}) &= \phi(i_V, z_1, \ldots, z_{v-1}, z_{v,m_v+1}, z_{v+1}, \ldots, z_{q_V}, y_V) \\
\phi'_v(z_{v,m_v}) &= \frac{\delta\phi(i_V, z_1, \ldots, z_{v-1}, z_{v,m_v}, z_{v+1}, \ldots, z_{q_V}, y_V)}{\delta z_v} \\
\phi'_v(z_{v,m_v+1}) &= \frac{\delta\phi(i_V, z_1, \ldots, z_{v-1}, z_{v,m_v+1}, z_{v+1}, \ldots, z_{q_V}, y_V)}{\delta z_v} \\
\phi'_n(z_{v,m_v}) &= \frac{\delta\phi(i_V, z_1, \ldots, z_{v-1}, z_{v,m_v}, z_{v+1}, \ldots, z_{q_V}, y_V)}{\delta z_n} \\
\phi'_n(z_{v,m_v+1}) &= \frac{\delta\phi(i_V, z_1, \ldots, z_{v-1}, z_{v,m_v+1}, z_{v+1}, \ldots, z_{q_V}, y_V)}{\delta z_n}
\end{aligned} \tag{6.34}$$

Here the function values have been interpolated using cubic spline interpolation while the derivatives are interpolated linearly since derivative data higher than the first derivatives is not available.

### 6.10.3 Symbolic Continuous Evidence

The updated potential table $\phi^*_{\mathcal{E}}(x_{V|Y_v=e})$ once the evidence $\mathcal{E} : Y_v = e$, for $v \in \Gamma_V$, has been entered is as follows:

$$\begin{aligned}
\phi^*_{\mathcal{E}}\left(x_{V|Y_v=e}\right) &= \phi^*_{\mathcal{E}}\left(i_V, z_V, y_1, \ldots, y_{v-1}, y_{v+1}, \ldots, y_{r_V}\right) \\
&= \left\{ \phi\left(i_V, z_V, y_1, \ldots, y_{v-1}, e, y_{v+1}, \ldots, y_{r_V}\right), \right. \\
&\quad \frac{\delta\phi\left(i_V, z_V, y_1, \ldots, y_{v-1}, e, y_{v+1}, \ldots, y_{r_V}\right)}{\delta z_1}, \ldots, \\
&\quad \left. \frac{\delta\phi\left(i_V, z_V, y_1, \ldots, y_{v-1}, e, y_{v+1}, \ldots, y_{r_V}\right)}{\delta z_{q_V}} \right\}
\end{aligned} \tag{6.35}$$

## 6.11 The Waste Incinerator Revisited

To illustrate the hybrid case we shall return to Lauritzen's waste incinerator problem (Lauritzen, 1992). The CPN in Figure 6.1 shows the independence graph of the associated model with variables 'Burning Regime' ($B$), 'Filter State' ($F$),

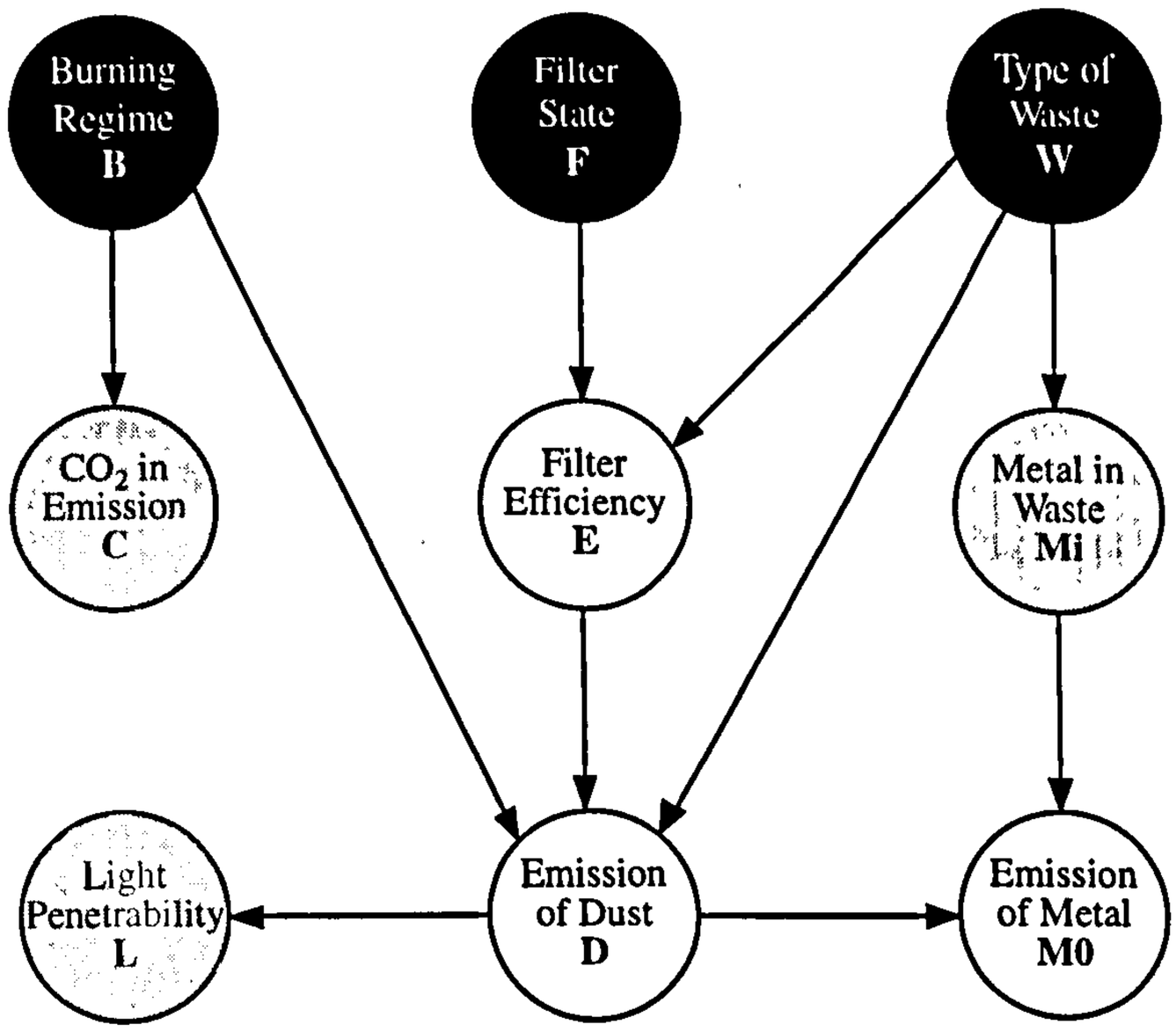



Figure 6.1: Causal probabilistic network for the waste incinerator problem. Discrete nodes are presented as black dots, spline interpolated continuous nodes as grey dots, and symbolic nodes as circles.

'Type of Waste' ($W$), '$CO_2$ in Emission' ($C$), 'Filter Efficiency' ($E$), 'Metal in Waste' ($Mi$), 'Light Penetrability' ($L$), 'Emission of Dust' ($D$), and 'Emission of Metal' ($M0$). The variables $B$, $F$, and $W$ are all discrete with states 'stable' or 'unstable', 'intact' or 'defective', and 'industrial' or 'household' respectively. The remaining variables $C$ , $D$ , $E$ , $L$, $Mi$, and $M0$ are continuous. Of these, $C$, $L$ and $Mi$ will be spline interpolated and $D$, $E$, and $M0$ will be represented symbolically. By construction the network contains both a spline interpolated node with a symbolic parent and a symbolic node with a spline interpolated parent. The junction tree corresponding to the graph of Figure 6.1 is given in Figure 6.2. This has been constructed by a process of moralisation and weak triangulation only. The cliques have been marked by ovals and the separators by rectangles. Note that the junction tree contains a mixture of both pure and hybrid universes.

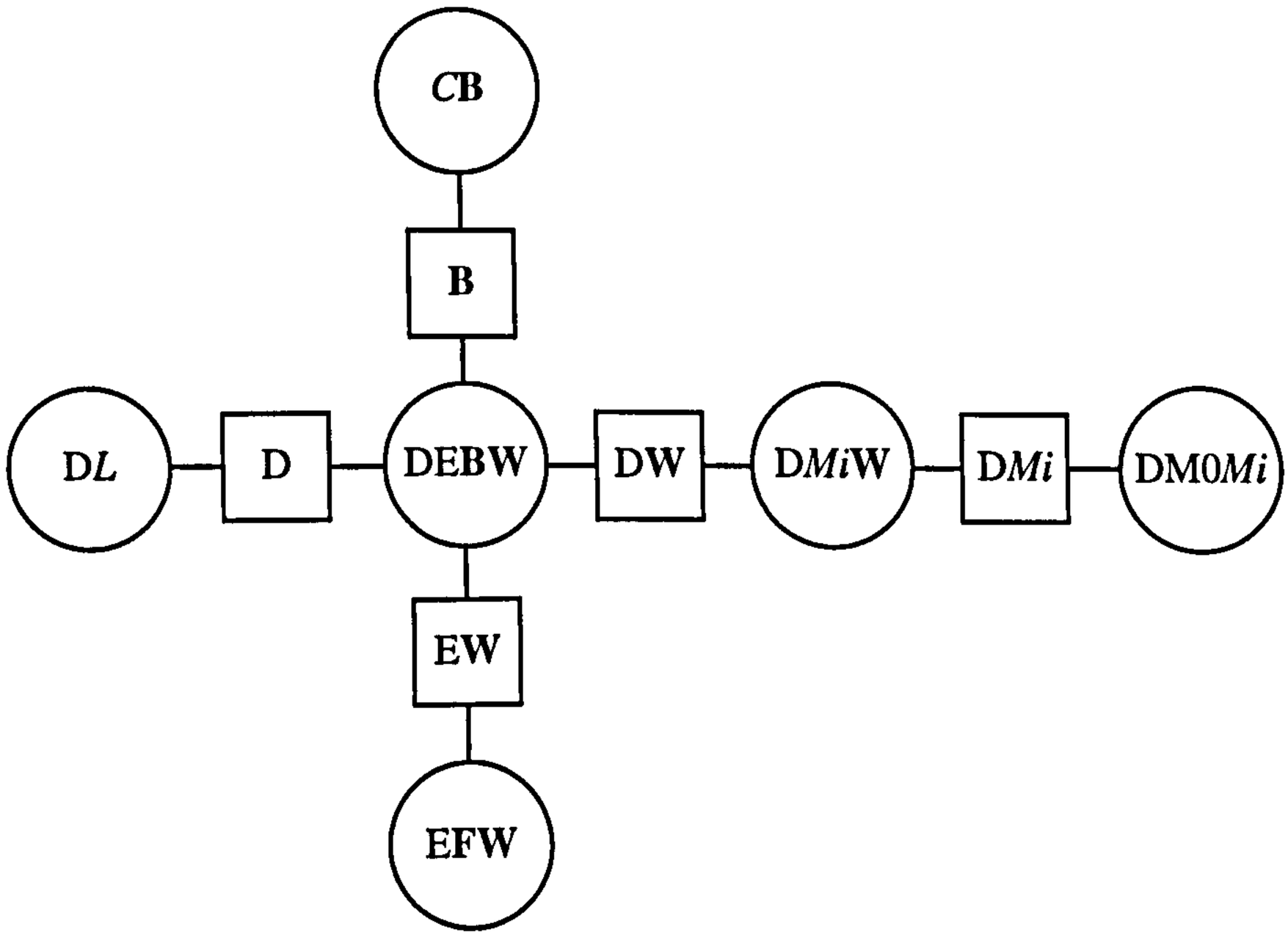



Figure 6.2: Junction tree for the waste incinerator problem. Discrete variables are presented in bold text, spline interpolated continuous variables as italic text, and symbolic variables as plain text.

The marginal probabilities of the discrete variables $B$, $F$ and $W$ and the conditional distributions of the continuous variables $C$, $D$, $E$, $L$, $Mi$ and $M0$ associated with the model are given in Table 4.5. The numbers of knots, knot sequence ranges, and knot widths of the three spline interpolated continuous variables $C$, $L$ and $Mi$ are those given in Table 5.1. It should be remembered that only the 9, 10 and 15 internal knots of $C$, $L$ and $Mi$, respectively, are actually needed to perform the

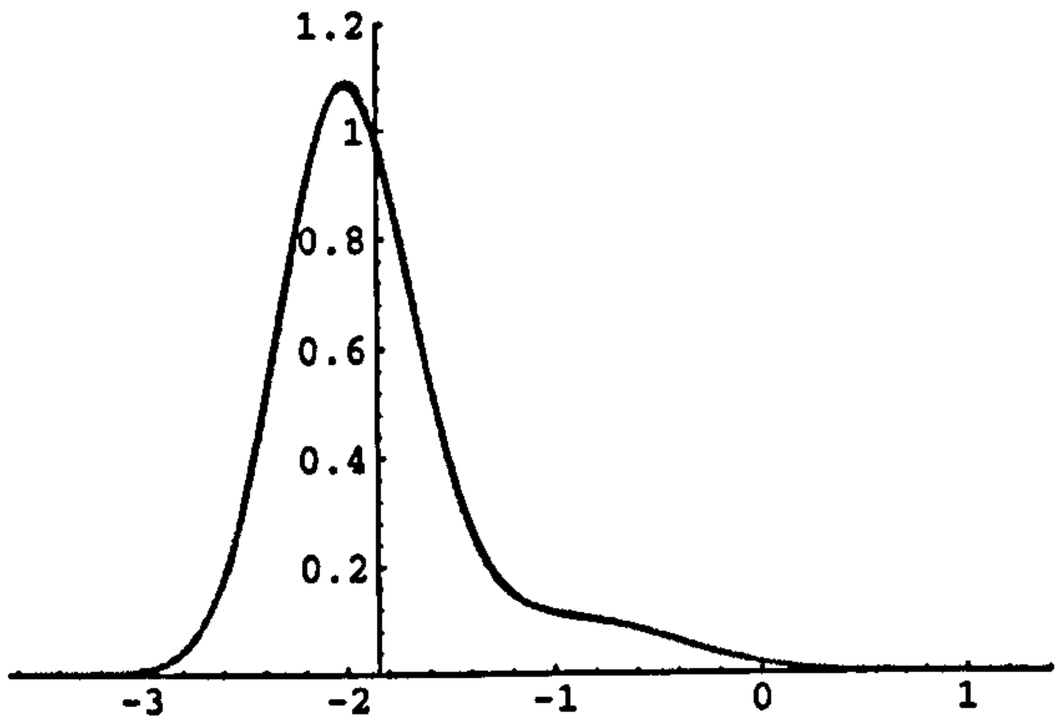
calculations. The choice of knots was made using the same techniques as those outlined in Chapter 5.

The potential tables for the marginal distributions of the discrete variables $B$, $F$, and $W$ consist of two numbers each. The potential table for the conditional distribution of $C \mid B$ has $(2 \times 9 \times 2)$ cells corresponding to the 2 levels of $B$, the 9 internal knots for $C$, and the need for both a function value and derivative with respect to $C$. This table consists of numbers only. Similarly the potential table for the conditional distribution of $Mi \mid W$ contains numeric values and has $(2 \times 15 \times 2)$ cells corresponding to the 2 levels of $W$, the 15 internal knots for $Mi$, and the function values and derivatives with respect to $Mi$. The potential tables for the remaining conditional distributions, $L \mid D$, $D \mid (B, E, W)$, $E \mid (F, W)$, and $M0 \mid (D, Mi)$, all contain symbolic equations for function values and, where applicable, derivatives. The potential table for $L \mid D$ has $(10 \times 2)$ cells corresponding to the 10 internal knots of $L$, and the symbolic function values and derivatives with respect to $L$. The potential tables for $D \mid (B, E, W)$ and $E \mid (F, W)$ are both $(2 \times 2)$ tables corresponding to the levels of their discrete components. The potential table for $M0 \mid (D, Mi)$ has $(15 \times 2)$ cells corresponding to the 15 internal knots of $Mi$, and the function values and derivatives with respect to $Mi$.

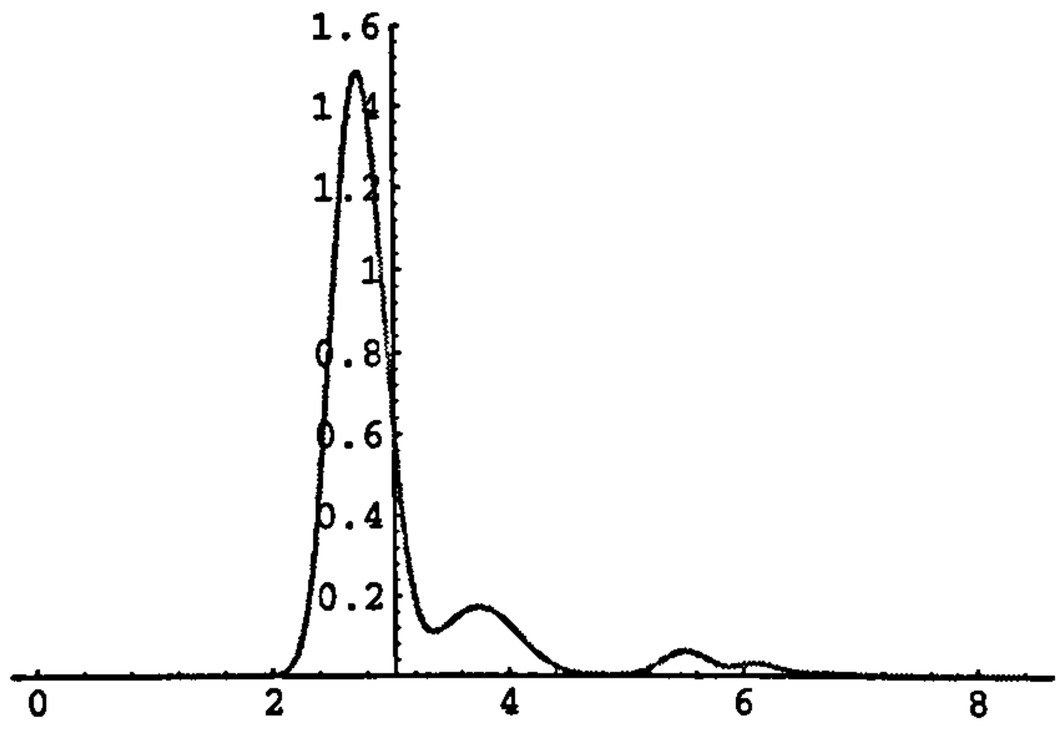
The system may be initialised using the random variable assignment given in Table 4.6. A suitable propagation schedule may then be passed to ensure that the potentials are consistent with each other. The propagation schedule given in Table 4.8 is appropriate for this. The results are presented in the next section.

## 6.12 Results

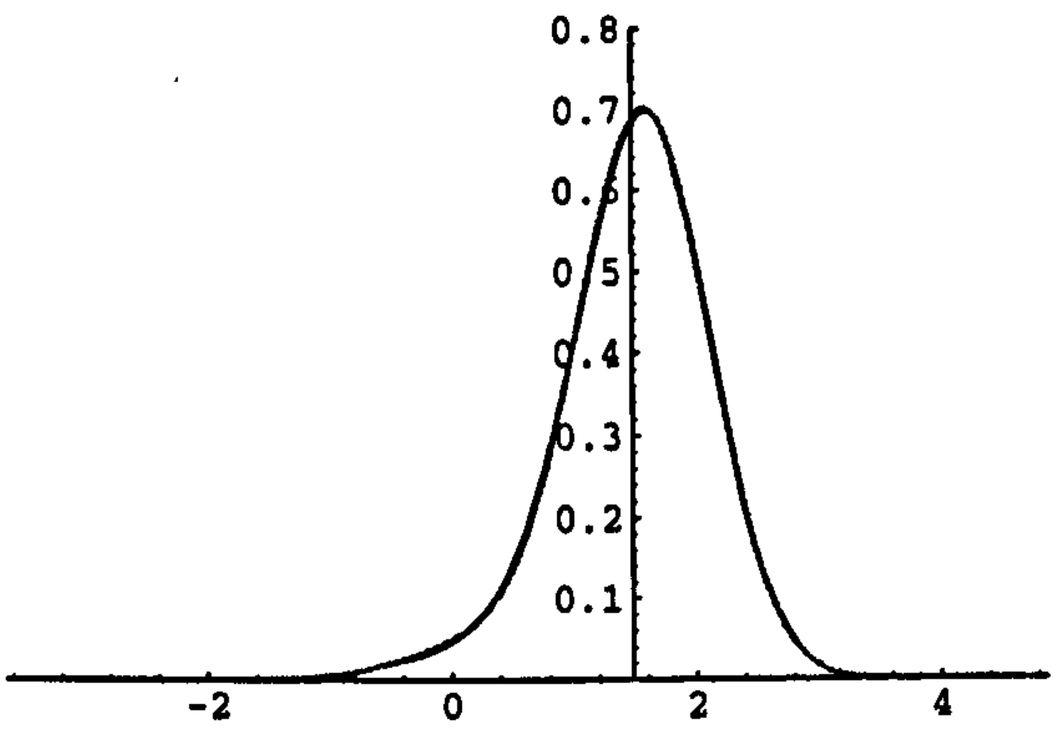
Following initialisation and the passage of a complete schedule of active flows the marginal distribution of any random variable $X_a$, for $a \in K$, may be determined from any universe containing that random variable by use of the marginalisation operator. Figure 6.3 presents graphs of the marginal distributions of the continuous variables in the Waste Incinerator problem in the absence of any evidence. The results obtained by use of the hybrid methodology described in this chapter are given in black. Results obtained using the exact methodology outlined in Chapter 4 are given in grey.



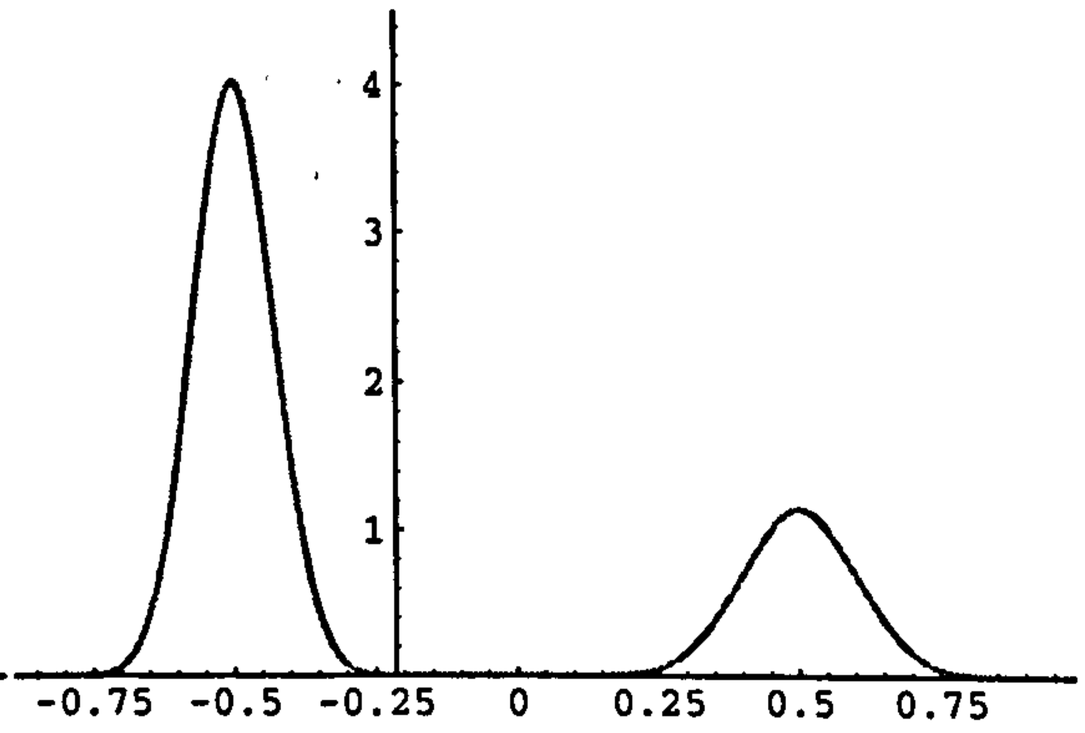

$C$ : $CO_2$ in emission



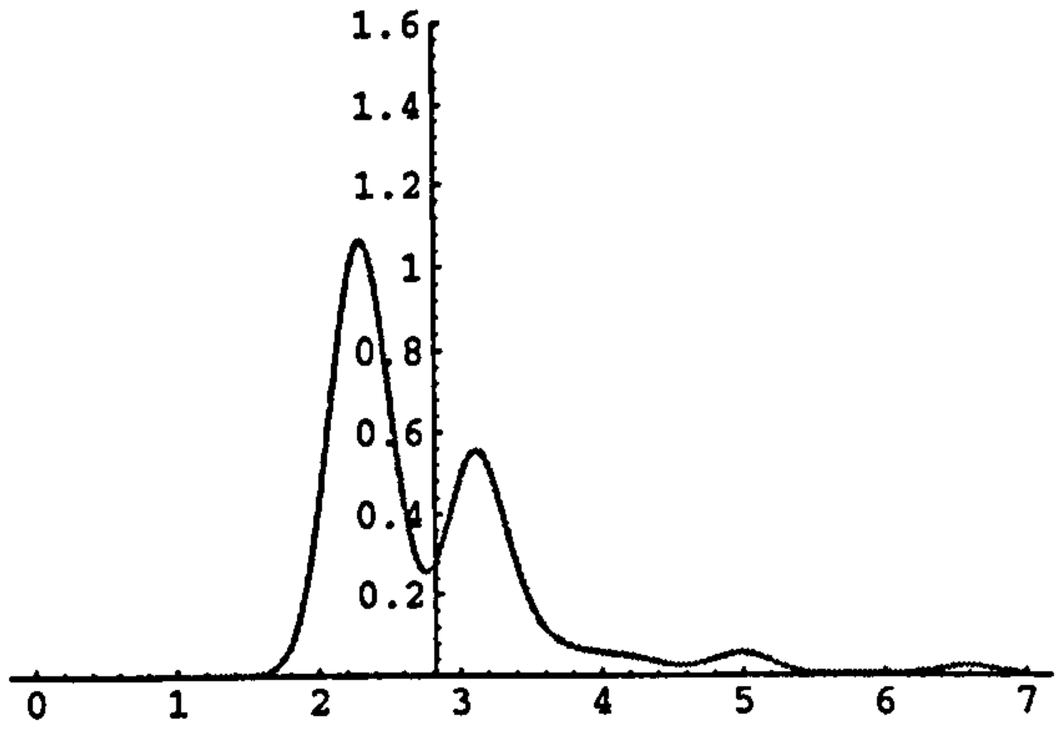

$D$ : Dust in emission




$L$ : Light penetrability




$Mi$ : Metal in waste




$M0$: Emission of metal

Figure 6.3: Graphs of the exact and hybrid marginal distributions of the continuous variables in the waste incinerator problem. Exact distributions are given in grey, hybrid ones in black.

The first two central moments of the marginal distributions of the spline interpolated and symbolic continuous variables may be determined by application of the techniques discussed in Sections 4.10.2 and 5.9. Table 6.1 presents the means and variances of the marginal distributions of the continuous random variables in the waste incinerator problem together with the probabilities of the discrete variables. The results of both the hybrid and exact methods are given.

| Variable | Type | Hybrid Method | Exact |
|---|---|---|---|
| B | Discrete | (0.85, 0.15) | (0.85, 0.15) |
| F | Discrete | (0.95, 0.05) | (0.95, 0.05) |
| W | Discrete | (0.285714, 0.714286) | (0.285714, 0.714286) |
| C | Spline | (-1.85032, 0.257314) | (-1.85, 0.2575) |
| L | Spline | (1.48064, 0.397735) | (1.48036, 0.398227) |
| Mi | Spline | (-0.214331, 0.214151) | (-0.214286, 0.21051) |
| D | Symbolic | (3.03929, 0.592909) | (3.03929, 0.592909) |
| E | Symbolic | (-3.25357, 0.502511) | (-3.25357, 0.502511) |
| M0 | Symbolic | (2.82494, 0.743742) | (2.825, 0.740113) |

Table 6.1: Means, variances and probabilities of the marginal distributions for the waste incinerator problem.

Figure 6.4 presents the graphs of the marginal distributions of the continuous variables in the presence of Lauritzen's evidence $\mathcal{E}$ : $\{W =$ Industrial, $C = -0.9$, and $L = 1.1\}$. The means, variances and probabilities of the marginal distributions given Lauritzen's evidence are given in Table 6.2.

Figure 6.5 presents the graphs of the marginal distributions of the continuous variables in the presence of Olesen's evidence $\mathcal{E}$ : $\{W =$ Industrial, $C = -1.6$, and $L = 0.5\}$. The means, variances and probabilities of the marginal distributions given Olesen's evidence are given in Table 6.3.

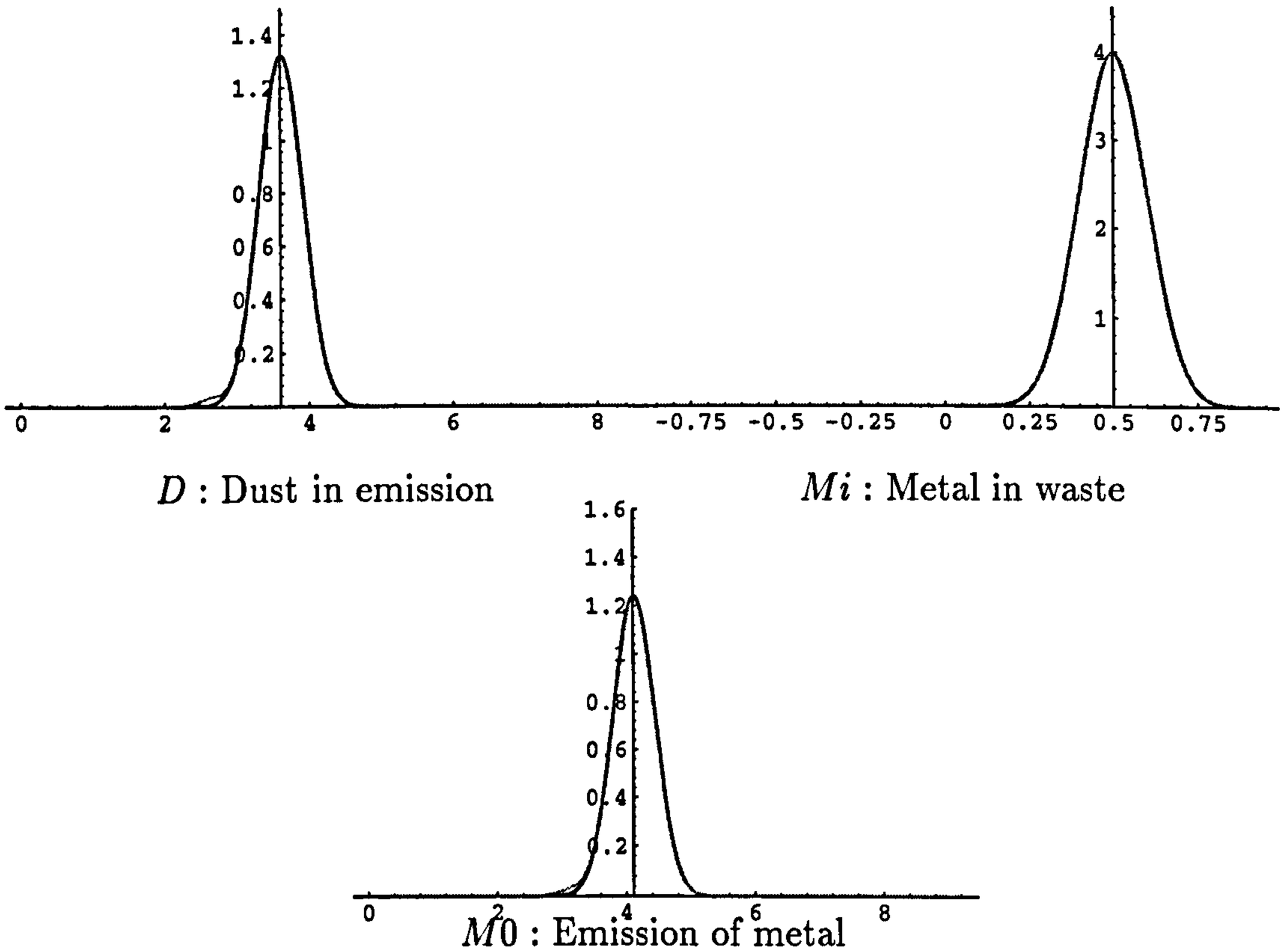



Figure 6.4: Graphs of the exact and hybrid marginal distributions of the continuous variables in the waste incinerator problem given Lauritzen's evidence ( $W =$ Industrial, $C = -0.9, L = 1.1$ ). Exact marginal distributions are given in grey and hybrid ones in black.

| Variable | Type | Hybrid Method | Exact |
|---|---|---|---|
| B | Discrete | (0.000128025, 0.999872) | (0.0122528, 0.987747) |
| F | Discrete | (0.999653, 0.000346644) | (0.999526, 0.000473728) |
| Mi | Spline | (0.500195, 0.0099399) | (0.5, 0.01) |
| D | Symbolic | (3.6191, 0.092647) | (3.60767, 0.106179) |
| E | Symbolic | (-3.89928, 0.00233593) | (-3.89834, 0.0058195) |
| M0 | Symbolic | (4.1194, 0.103591) | (4.10767, 0.118179) |

Table 6.2: Means, variances and probabilities of the marginal distributions for the waste incinerator problem given Lauritzen's evidence ( $W =$ Industrial, $C = -0.9, L = 1.1$ ).

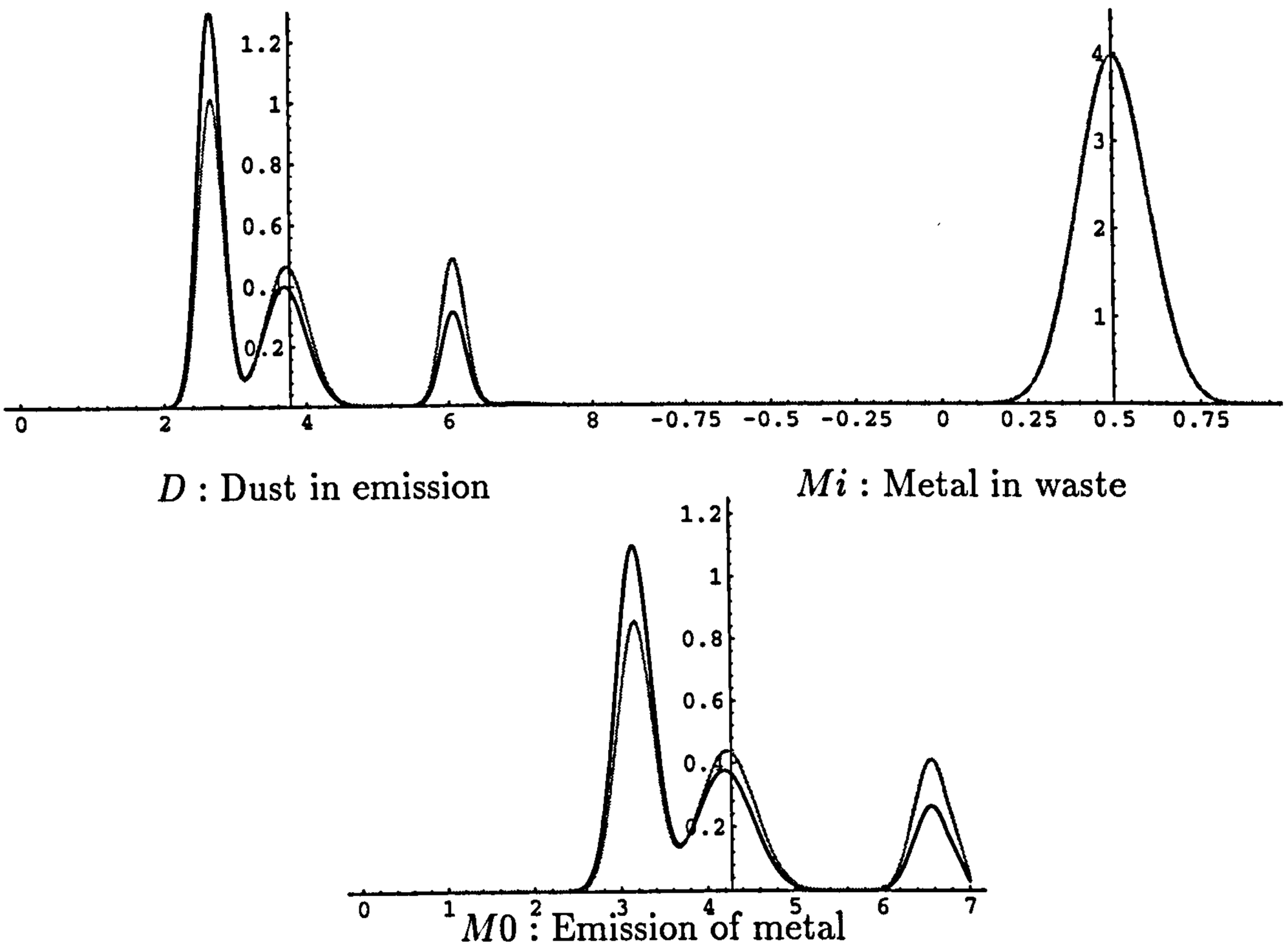



Figure 6.5: Graphs of the exact and hybrid marginal distributions of the continuous variables in the waste incinerator problem given Olesen's evidence ( $W$ = Industrial, $C = -1.6, L = 0.5$ ). Exact marginal distributions are given in grey and hybrid ones in black.

| Variable | Type | Hybrid Method | Exact |
|---|---|---|---|
| B | Discrete | (0.691439, 0.308561) | (0.642434, 0.357566) |
| F | Discrete | (0.862364, 0.137636) | (0.785816, 0.214184) |
| Mi | Spline | (0.500106, 0.00998236) | (0.5, 0.01) |
| D | Symbolic | (3.44747, 1.37972) | (3.77448, 1.73616) |
| E | Symbolic | (-3.41826, 1.45389) | (-3.15035, 2.06165) |
| M0 | Symbolic | (3.94766, 1.39165) | (4.27448, 1.74816) |

Table 6.3: Means, variances and probabilities of the marginal distributions for the waste incinerator problem given Olesen's evidence ( $W$ = Industrial, $C = -1.6, L = 0.5$ ).

## 6.13 Conclusions

This chapter has shown how the techniques developed in previous chapters may be united to form a single, hybrid, framework. In particular we have shown how discrete exact, symbolic, and spline interpolated random variables may coexist not only within the same junction tree but also within the same universes of that junction tree. The key distinction between this approach and that taken by Dawid *et al.* (1993) is the level at which hybrid techniques may be employed. Our approach ties the computational technique to the random variable resulting in the formation of hybrid universes. The approach taken by Dawid *et al.* ties the computational technique to the universe producing hybrid trees.

The hybrid tree approach will, in general, allow the inclusion of any suitable computational method (discrete exact, spline interpolated, symbolic, monte carlo, numeric integration, and so on). This feature makes for an elegant object orientated approach with a clear confinement of technical difficulties to the universe level, rather than that of the variable. While intra-universe computations are simplified by this approach, inter-universe computations may be more difficult to perform. Consider, for example, neighbouring universes which try to employ symbolic and monte carlo methods. Either one has to deal with computations between these two universes using some, possibly *ad hoc*, methodology or constraints must be put on the form of inter-universe computation which may occur. For example one could constrain the interaction between the two universes to some purely numeric relationship. The problems introduction by inter-universe computations may either cause an increase in universe size or force choices as to which methods may be employed in which universes.

In contrast the hybrid universe approach makes no demands to increase universe size requiring only a weakly triangulated graph upon which to build its junction tree. Both inter and intra-universe computations require a consistent methodology. If the hybrid universe approach is applied with the constraints on the interaction between Gaussian and non-Gaussian variables as described in Section 6.7 then it is fully implementable using only the methodology we developed in this chapter. The cost of such an approach may be seen in the increased complexity of the computations required. This may be particularly apparent in universes which inherit large functions formed as a result of the interaction between symbolic and spline variables.

If, for the waste incinerator problem in the absence of evidence, we compare the results generated by the implementation of a hybrid methodology with the exact results generated by the use of symbolic techniques we observe a high level of accuracy. This close comparison may be observed in the probability densities, means and variances of the marginal distributions of the continuous random variables, and also in the marginal probabilities of the discrete random variables. However, a noticeable difference in the level of accuracy may be observed following the inclusion of evidence. Although such inaccuracies are slight in the case of Lauritzen's evidence, they are more extreme in the case of Olesen's evidence. Even if such accuracies may be deemed acceptable or even unavoidable it is still important to understand from where they are derived.

The answer to this problem lies in the entry of continuous evidence on the spline interpolated variables (in our examples on $C$ and $L$). The accuracy with which evidence may be added will be determined by a number of different factors yet all these factors relate to the location of the evidence on the joint probability density function. Recall that a different approach to evidence entry must be taken if that evidence falls on a knot or not. If, for a piece of evidence on a single random variable, evidence falls on a knot then, in the absence of any other evidence, evidence entry will be exact. If, however, that evidence falls between two knots then interpolation is required to enter that evidence. The closer the evidence is to one of the knots the more accurate evidence entry will be. As a consequence of this one way of increasing accuracy in the presence of evidence might be to increase the number of knots. Doubling the number of knots would shrink the knot width by half and would ensure that a piece of evidence were either as close to, or closer to a knot than it was before. The "cheat's method" would be to rebuild the interpolation lattice according to the observed collection of evidence to ensure that the evidence fell on an interpolation node and was thus entered exactly.

Not only will accuracy depend upon the proximity of a piece of evidence to a knot, but it will also depend upon which part of the probability density surface it falls. This is because when splines are fitted accuracy tends to be measured in absolute terms. It is, however, accuracy in relative terms which will have an impact when adding evidence. Hence the more extreme a collection of evidence the less accurate the results are likely to be. An extreme collection of evidence will be one with a relatively low normalisation constant. Note that the normalisation

constant for the addition of Lauritzen's evidence is 0.022066 compared with only 0.014452 for the addition of Olesen's evidence. This latter normalisation constant is still quite moderate. Suppose we were to observe some collection of evidence on the periphery of the probability density surface - here the normalisation constant will be very low and hence very difficult to model well in relative terms. This could result in huge discrepancies between the approximated and true results. Evidence entry should therefore be viewed with some caution in such circumstances. The most extreme collection of evidence would be one which fell outside of the interpolation lattice. The realisation of such a collection of evidence would lead us to either redefine our model, or to reject that evidence as data errors.

# Chapter 7

# Conclusions

The primary objectives of this work were to develop techniques to facilitate the construction of PESs and to improve their useability. PESs provide a method for the specification and handling of the joint distribution of a finite set of random variables. They employ graph theory to decompose a large multivariate problem into a series of smaller interrelated multivariate problems, an approach which both structures and simplifies the solution.

The techniques we have developed here are in keeping with this philosophy. We have extended the range of random variables which may be incorporated into a PES by the introduction of symbolic and spline methods. In particular, this work expands on the current literature's treatment of continuous random variables. Symbolic methods enable random variables to be modelled exactly through their probability density functions. This provides a modelling environment which has no loss of information in contrast to previous techniques which will only allow the handling of moments. It is also the most natural approach to PES construction which makes it compatible with other methodologies adopting the probability density function approach.

While symbolic techniques have sought to make the modelling of well behaved distributions computationally feasible, spline techniques have sought to open up the probability density function approach to less well behaved distributions. Splines simplify the application of the most costly operator required to implement a PES - that of marginalisation. The marginalisation of a spline interpolated potential function with respect to a continuous random variable becomes simply an extension of discrete exact marginalisation.

The probability density function approach to PES construction via the use

of splines and symbolics has the advantage that it may be implemented using a weakly triangulated network. This will, in general, enable the use of universes which are as small as possible - one of the key objectives of PESs.

Handling probability density functions rather than a set of moments which provide an insufficient parameterisation of a joint distribution, enables a greater understanding to be gained of a multivariate system. A PES is supposed to encapsulate an expert's knowledge. To maximise its usefulness the output it produces should be readily interpretable by expert and non-expert alike. A symbolic PES provides all the information required to plot joint or marginal distributions of sets of random variables. The incorporation of symbolic evidence provides a methodology by which one variable's effect on the other variables in a system may be better understood. A spline interpolated PES provides all the information required to plot the marginal distributions of the random variables it comprises. Such advantages make the phrase "a picture paints a thousand words" ring true.

Which method should be employed in which PES will sometimes be a matter of choice. Ideally it is the most accurate method which should be implemented. This will tend to suggest that symbolic techniques are, in absence of any theoretical constraints, the most appropriate. In some cases, however, where full information is not required, numeric techniques may serve as a suitable replacement. Additionally there may be a compromise to be made due to computational feasibility. A method may be deemed infeasible for theoretical reasons. For example, there may be problems in deriving a closed form solution. A goal of this thesis has been to make the theoretically feasible feasible. The techniques developed, however, are not necessarily feasible in practice. They may, for example, be too computationally expensive to implement. Even in the discrete exact case computational problems can arise. Recall, for example, that a universe comprising just twenty binary variables will have over a million possible states.

In general, the complexity of the underlying calculations required to tackle a problem will be reflected in both the size and complexity of the underlying data structures and the time taken to perform calculations. For the waste incinerator example Table 7.1 lists the times taken to:

a) Perform preprocessing tasks: Initialise a PES, add evidence and pass a propagation schedule.

b) Perform postprocessing tasks: Determine the probabilities, mean and variance of the marginal distributions of every random variable in the system and, where appropriate, determine and plot the marginal probability density function.

Timings are given for Lauritzen's numeric approach (Chapter 3), the exact symbolic approach (Chapter 4), the spline approach (Chapter 5), and the hybrid approach (Chapter 6). It should be noted that Lauritzen's approach is given an unfair advantage in the fact that marginal probability density functions are neither derived nor plotted. Similarly for the spline and hybrid approaches no timings are presented for the generation of appropriate interpolation lattices. These omissions should be considered when passing any judgements on the results.

| Method | Evidence | Preprocessing Time | Postprocessing Time | Total Time |
|---|---|---|---|---|
| Numeric | None | 35.43s | 18.81s | 54.24s |
| | Lautitzen's | 31.16s | 10.65s | 41.81s |
| | Olesen's | 31.18s | 10.73s | 41.91s |
| Exact | None | 41.17s | 710.05s | 751.22s |
| | Lauritzen's | 34.44s | 533.80s | 568.24s |
| | Olesen's | 33.97s | 561.89s | 595.86s |
| Spline | None | 790.51s | 339.22s | 1129.73s |
| | Lauritzen's | 744.42s | 274.03s | 1018.45s |
| | Olesen's | 747.73s | 276.77s | 1024.50s |
| Hybrid | None | 73.51s | 216.43s | 582.36s |
| | Lauritzen's | 635.85s | 1280.11s | 1915.96s |
| | Olesen's | 3949.05s | 7480.71s | 11429.76s |

Table 7.1: A comparison of the computation speeds of the four propagation methods outlined in this work.

The times presented should be viewed as relative guides to the computational speeds of the four methods. All the examples were implemented using *Mathematica 2.0 for Students* (a slower version of *Mathematica 2.0*) on low-end Macintosh systems. Displayed times were those achieved on a Macintosh Performa 6200

with 16Mb of memory and a Power PC chip operating at 75Mhz. The same calculations were achievable in four times the time on a (five year older) Macintosh LC with 10Mb of memory and a 68020 chip operating at 16Mhz. While such considerations may be viewed as inconsequential - the relative speed taken to perform the various methods being far more important than the relative speed of different computers - they do bear out the continuing increase in computer power. Computers are becoming more and more powerful as processors get faster and memory gets cheaper. These advances promise to improve the feasibility of computational techniques. This should, however, not be seen as an excuse for the creation, or use, of less efficient algorithms but it will certainly open up new possibilities for PESs both in terms of the size and complexity of problems which can be tackled. If anything this will demand the application of robust methods.

While this work has concentrated on the introduction of symbolic and spline techniques into PESs, it has by no means exhausted these subjects. Our symbolic techniques explored the representation of Gaussian random variables and discrete random variables of infinite or indeterminate state space. We also discussed the inclusion of symbolic parameters into PESs. While they are most useful and enabled a description of the basic foundations of computer algebra, pattern matching, and simplification techniques, the cases we have covered are, of course, limited. Techniques should be developed to extend these approaches to other distributional forms. One problem area we noted was that which stems from the marginalisation of a potential function, with respect to a continuous random variable, which has no closed form solution. We solved this problem by introducing the spline interpolation of continuous random variables. What should one do, however, if one is faced with a problematic summation of a potential function with respect to a discrete random variable? This is an area requiring further investigation. While this work concentrated on the handling of continuous random variables, perhaps an equivalent work is required to explore the handling of discrete random variables which are not of the discrete exact case.

We showed how splines may be used to represent continuous distributions in PESs. The particular propagation algorithm we implemented assumed that these splines had equally spaced knots. This approach has the desirable advantage that the order with which one marginalises has no impact on the result. In some cases the reliance on equally spaced knots may prove to be an unacceptable prerequisite. Consider, for example, a mixture distribution comprising two Gaussian random

variables. Let one have a very small variance in comparison to the other, and assume that their means are relatively far apart. To fit an equally spaced spline to this distribution with any accuracy would require a knot width small enough to satisfactorily capture the shape of the underlying distribution with the lower variance. This constraint will imply that a very large number of knots will be required since the full range spanned by the two underlying distributions will be big in comparison to the required knot width. A more appropriate approach might be to concentrate knots around the means of the two underlying distributions the spread, number and knot width, of these knots being determined by their variances. This would result in a reduced number of knots with varying knot width. This would, in turn, improve computational feasibility with, perhaps, no loss in accuracy. Methods to sensibly implement such alterations should be investigated.

In general the approximate results derived from a spline interpolated PES will approach the true results for that PES if one increases the number of knots in the system. The more knots that are added to a system, however, the bigger the computational problem. There is therefore a trade-off between accuracy, size and speed. To ensure the accuracy of a PES in the absence of evidence one can simply compare the univariate conditional distributions with their spline interpolations. How can one improve, or guarantee, accuracy when evidence is added? In our examples we compared exact results with approximate spline interpolated results to make a call over accuracy. In the "real world" when spline methods will probably be employed only because no exact method is appropriate we can not tell what degree of accuracy we have achieved. A solution to this may be the "cheat's method" which redefines the interpolation lattice according to received evidence. This may be inappropriate in large systems where a significant cost may be involved in rebuilding that lattice from scratch. Throwing in an extra point on the observed variables without readjusting the lattice on the unobserved variables may be acceptable and is certainly an area worthy of future research.

Cubic spline interpolation effectively discretises a network into a set of points and slopes at those points. Investigations should be carried out into the relative merits of this approach and a linear spline approach which would need to employ no derivative information. This latter approach would effectively view a continuous distribution as a discrete exact random variable. Savings from the absence of derivative data may be turned into an increased number of knots. When would

this outweigh the value of the derivative data?

The extension and improvement of other numeric techniques should also not be overlooked. We added simulation techniques to Lauritzen's numeric handling of the means, variances, and probabilities in mixed graphical association models. This enabled the generation of approximate probability density functions for the marginal distributions of the continuous random variables using kernel density estimation. In turn this provided a graphical way in which the non-expert may better interpret those marginal distributions using more than just their means and variances.

The basic symbolic methodology we developed to model this framework is one which can robustly be adapted to suit purely numeric calculations. This requires the mean vectors, covariance matrices, and weights to be stored for each term in a CGM-distribution. In contrast to Lauritzen's scheme only a weakly triangulated network is required reducing universe size, and there is no loss of information. The cost of this approaches that the data structures concerned are not fixed in size and may grow on evidence entry due to the modelling of CGM-distributions rather than CG-distributions acting as a proxy for the calculation of means and variances. In contrast to symbolic techniques this approach would not allow the inclusion of symbolic evidence (if implemented purely numerically), and would be harder to reconcile with other techniques such as spline methods. It may, however, be faster and more space efficient.

In conclusion, while this thesis has begun to answer a number of questions regarding how PESs may be extended to model new variable sets the answers it has provided are by no means exhaustive and this area is one which remains fertile for future research.

# References


Aitken, C. G. G. & Gammerman, A. (1989). Probabilistic reasoning in evidential assessment. *Journal of Forensic Science Society*, **29**, 303–316.

Aitken, C. G. G., Gammerman, A., Zhang, G., Connolly, T., Baily, D., Gordon, R., & Oldfield, R. (1996). Bayesian belief networks with application in specific case analysis. In *Computational learning and probabilistic reasoning*. Wiley and Sons.

Andreassen, S., Jensen, F. V., Andersen, S. K., Falck, B., Kjærulff, U., Woldbye, M., Sørensen, A., Rosenfalck, A., & Jensen, F. (1989). MUNIN - an expert EMG assistant. In Desmedt, J. E. (Ed.), *Computer-aided electromyography and expert systems*, pp. 255–277. Elsevier Science, Amsterdam, Holland.

D'Ambrosio, B. (1994). Algebraic techniques for efficient inference in Bayesian networks. Tech. rep., Department of Computer Science, Oregon State University.

Dawid, A., Kjærulff, U., & Lauritzen, S. L. (1993). Hybrid propagation in junction trees. Tech. rep. R-93-2028, Institute for Electronic Systems, Aalborg University, Denmark.

Dawid, A. P. (1979a). *Conditional independence in statistical theory*, Vol. 41 of *B*, pp. 1–31. Journal of the Royal Statistical Society.

Dawid, A. P. (1979b). *Some misleading arguments involving conditional independence*, Vol. 41 of *B*, pp. 249–252. Journal of the Royal Statistical Society.

Dawid, A. P. (1992). Applications of a general propagation algorithm for probabilistic expert systems. *Statistics in Computing*, **2**, 25–36.

Franklin, R., Spiegelhalter, D., Macartney, F., & Bull, K. (1989). Combining clinical judgements and statistical data in expert systems: over the telephone

management decisions for critical congenital heart disease in the first month of life. *International Journal of Clinical Monitoring and Computing*, **6**, 157–166.

Frydenberg, M. & Lauritzen, S. L. (1989). Decomposition of maximum likelihood in mixed graphical interaction models. *Biometrika*, **76**, 539–555.

Frydenberg, M. (1989). The chain graph markov property. Tech. rep. 186, Department Theory Statistics, Aarhus University, Denmark.

Gammerman, A. J., Aitken, C. G. G., Luo, Z., & Talbot, M. (1991). A computational system for mixed probabilistic models. *Statistics in Computing*, **1**.

Geweke, J. (1991). Generic, algorithmic approaches to monte carlo integration in Bayesian inference. In Flournoy, N. & Tsutakawa, R. K. (Eds.), *Statistical Multiple Integration.* American Mathematical Society.

Golumbic, M. C. (1980). *Graph Theory and Perfect Graphs.* Academic Press, New York.

Hall, C. (1979). Spline blended approximation of multivariate functions. In Sahney, B. (Ed.), *Polynomial and Spline Approximation, Theory and Applications.* D. Reidal Publishing Company, Dordrecht, Holland.

Heckerman, D., Horvitz, E., & Nathwani, B. (1992). Towards normative expert systems: Part i. the pathfinder project. *Methods of Information in Medicine*, **31**, 90–105.

Heckerman, D., Breese, J., & Rommelse, K. (1995). Decision-theoretic troubleshooting. *Communications of the ACM*, *38*(3), 49–56.

Horvitz, E. & Barry, B. (1995). Display of information for time-critical decision making. In Besnard & Hanks (Eds.), *Proceedings of the eleventh conference on uncertainty in artificial intelligence*, pp. 296–305, San Fransisco, CA. Morgan Kaufmann.

Jensen, F. V., Lauritzen, S. L., & Olesen, K. G. (1990). Bayesian updating in causal probabilistic networks by local computations. *Computational Statistics Quarterly*, 4, 269–282.

Jensen, F. V. (1996). *An introduction to Bayesian networks.* UCL Press, London, England.

Johnson, S. C. (1975). Yacc: yet another compiler-compiler. Tech. rep. 32, Bell Laboratories, Murray Hill,New Jersey.

Kjærulff, U. (1990). Triangulation of graphs- algorithms giving small total state space. Tech. rep. R 90-09, Institute for Electronic Systems, Aalborg University, Denmark.

Lancaster, P. & Salkauskas, K. (1986). *Curve and surface fitting: An introduction.* Academic Press Inc, London.

Lauritzen, S. L. & Spiegelhalter, D. G. (1988). *Local computations with probabilities on graphical structures and their application to expert systems (with discussion)*, Vol. 50 of *B*, pp. 157–224. Journal of the Royal Statistical Society.

Lauritzen, S. L. & Wermuth, N. (1984). Mixed interaction models. Tech. rep. R 84-08, Institute for Electronic Systems, Aalborg University, Denmark.

Lauritzen, S. L. & Wermuth, N. (1989). Graphical models for associations between variables, some of which are qualitative and some quantitative. *Annals of Statistics*, **17**, 31–57.

Lauritzen, S. L., Dawid, A., Larsen, B. N., & Leimer, H. G. (1990). Independence properties of directed markov fields. *Networks*, **20**, 491–505.

Lauritzen, S. L. (1989). Mixed graphical association models. *Scandinavian Journal of Statistics*, **16**, 273–306.

Lauritzen, S. L. (1992). Propagation of probabilities, means and variances in mixed graphical association models. *Journal of the American Statistical Association*, **86**, 1098–1108.

Leimer, H. G. (1989). Triangulated graphs with marked vertices. in graph theory in memory of g.a. dirac (eds l.d. andersen *et al.*). *Annals of Discrete Mathematics*, **41**, 311–324.

Lesk, M. E. (1975). Lex: a lexical analyser generator. Tech. rep. 39, Bell Laboratories, Murray Hill,New Jersey.

Li, Z. & D'Ambrosio, B. (1994). Efficient inference in Bayes networks as a combinatorial problem. *International Journal of Approximate Reasoning*, **11**.

Ng, K. C. & Abramson, B. (1990). Uncertainty management in expert systems. *IEEE Expert Systems*, **April**, 29–48.

Olesen, K. (1991). Causal probabilistic networks with both discrete and continuous variables. Tech. rep. R 91-29, Institute for Electronic Systems, Aalborg University, Denmark.

Pearl, J. (1982). Reverend Bayes on inference engines: a distributed hierarchical approach. In *Proceedings of American Association for Artificial Intelligence National Conference on AI, Pittsburgh*, pp. 133–136.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference.* Morgan Kaufmann, San Mateo, California.

Rasmussen, L. K. (1995). Bayesian network for blood typing and parentage verification of cattle. Tech. rep. 38, Aalborg University, Department of Mathematics and Computer Science.

Spiegelhalter, D. J. (1990). Fast algorithms for probabilistic reasoning in influence diagrams, with applications in genetics and expert systems. In Oliver, R. M. & Smith, J. Q. (Eds.), *Influence diagrams, belief nets and decision analysis*, pp. 361–384, John Wiley and Sons Ltd., Chichester.

Wermuth, N. & Lauritzen, S. L. (1989). *On substantive research hypotheses, conditional independence graphs and graphical chain models (with discussion)*, Vol. 51 of *B.* Journal of the Royal Statistical Society.

Whittaker, J. (1990). *Graphical models in applied multivariate statistics.* John Wiley and Sons, Chichester.

Wolfram, S. (1991). Mathematica*: a system for doing mathematics by computer (2nd Edition).* Addison-Wesley Publishing Co., Redwood City, California.